Microsoft® SharePoint® Server 2010

Microsoft SharePoint Server 2010

Evaluation Guide

Microsoft®

Copyright

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, Access, Active Directory, InfoPath, Internet Explorer, MS-DOS, Outlook, PowerPoint, SharePoint, Visual Studio, Windows, Windows PowerShell, and Windows Server are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

Abstract 1
Introduction 1
Summary of What's New 3
Governance 4
High Availability 4
Deployment and Upgrade 4
System Requirements 4
Installation 5
Upgrade from Office SharePoint Server 2007 6
Preparing for Upgrade 6
Upgrade Methods 7
Downtime Mitigation 8
Visual Upgrade 8
Upgrade Logging 9
Upgrade Scenario 9
Patch Management Improvements 11
SharePoint Central Administration Web Site 12
Service Applications 15
Search 17
Search Service 18
Search Architecture 18
Crawling Architecture 18
Improving Search Performance 20
FAST Search for SharePoint 20
Business Connectivity Services 21
Business Data Connectivity Service Application and Model 21

External Content Types 22
SharePoint Server and Office Integration 22
Managed Metadata Service 23
Managed Metadata Service Application 23
Managed Metadata Service Connection 24
Example Scenario 25
Claims-based Authentication 25
Service Applications Conclusion 26
Health and Monitoring 26
Diagnostics 28
Reliability and Monitoring 30
Remote BLOB Storage 31
Performance Controls 31
Throttling 31
Controlling Large List Activities 32
Windows PowerShell Administration 34
Cmdlets 34
Scenario: Provisioning of Managed Accounts 37
Windows PowerShell Out-Performing Stsadm Scenario 38
Managed Accounts 39
Backup and Restore 40
Central Administration vs. Windows PowerShell 40
Granular Backup and Restore 41
Back Up a Site Collection 41
Export a Site or List 42
Recover Data from an Unattached Content Database 43
Scenario: Using Windows PowerShell to Script Backing Up All Site Collections Individually 45

Farm Backup and Restore 45
Read-Only Databases 45
SQL Mirroring 46
SQL Snapshots 47
Search and Index Backup and Restore 47
Multi-tenancy and Hosting 47
Site Subscriptions 47
Administration 48
Branding 49
Themes 49
SharePoint Designer 50
Master Pages 51
Silverlight 52
Conclusion 52

Abstract

This evaluation guide is designed to give the IT pro an introduction and overview of the features in Microsoft® SharePoint® Server 2010 Beta that are most pertinent to installing, managing, and configuring the SharePoint farm. It begins with a brief introduction and a summary of what's new, followed by a more in-depth discussion of the most relevant features.

The ultimate goal of this guide is to provide the IT pro with the understanding necessary for installing and evaluating SharePoint Server 2010. This guide is intended for the Windows Server® administrator, Office SharePoint Server 2007 administrator, Web server administrator, or any IT pro involved in server administration.

This evaluation guide is subject to change. For the latest information about SharePoint Server 2010, go to http://www.microsoft.com/sharepoint.

Introduction

Welcome to the IT pro's *Microsoft® SharePoint® Server 2010 Evaluation Guide*. The goal of this guide is to help you gain sufficient knowledge and understanding to evaluate this product from Microsoft.

SharePoint Server 2010 is the business collaboration platform for the enterprise and the Web. The platform empowers people and enables them to connect through an integrated set of rich features. This common collaboration environment provides a unified infrastructure, which helps organizations cut costs while allowing them to rapidly respond to business needs. SharePoint Server 2010 can be deployed onsite (also called on-premises) or as hosted services, and can be deployed by using virtualization technology or live instances. Whether on-premises or hosted, deploying SharePoint Server requires the IT pro to be heavily involved. Microsoft SharePoint Server 2010 includes a large number of improvements to features that the administrator is familiar with, and it also includes many new features.

Specifically, SharePoint Server 2010 provides the IT pro with:

- **IT pro productivity** by improving the administrative experience and by giving the administrator deeper operational insight. This increase in productivity is

facilitated by a new, streamlined Central Administration Web site, new capabilities to manage and monitor the SharePoint farm, and Windows PowerShell™ support, just to name a few features.

- **Scalable unified infrastructure** that includes better control over server resources (for example, to improve the performance and management of large lists) and data management and protection by using high availability. We have also introduced a new scalable services architecture that allows you to effectively manage and centralize services like Search, My Sites, and Taxonomy. These service applications can be managed through Central Administration, and they can be managed and scripted by using Windows PowerShell as well. The new services architecture is extensible, so third-party companies can build services and add them to the platform.
- **Flexible deployment** by providing a quick and simple installation and configuration process. The upgrade from Office SharePoint Server 2007 is smooth and predictable. We've also introduced functionality that allows you to manage how your users use SharePoint. For example, you can now safely allow site administrators to upload custom user code by using Sandboxed Solutions. You also have more governance control: If you want to maintain a centralized SharePoint deployment versus a more decentralized approach, you can either block those deployments through Group Policy or track them by using Active Directory® marker support.

SharePoint Server 2010 provides a very large number of features, and a detailed accounting of each feature is well beyond the scope of this document. This guide will describe some of the most compelling improvements and new additions to the SharePoint Server 2010 platform for IT pros. With this evaluation guide, the IT pro will be able to properly evaluate these new features and readily describe their capabilities to their colleagues, clients, and business partners.

Next, we will briefly discuss a summary of the new features, followed by a brief summary of how these features can be used to achieve functions such as governance and high availability. Then we will move on to the details of each of the pertinent features.

Summary of What's New

Microsoft® SharePoint® Server 2010 includes numerous new capabilities and features only some of which are highlighted in this guide. Here is a brief summary of what's new.

IT pro productivity
SharePoint Best Practices Analyzer
Backup and restore configuration data
Site collection migration
Content recovery down to the site and list level
Content restore of site or list
Windows PowerShell™ snap-ins and commands
Scalable unified infrastructure
Shared service applications
Hosting and multi-tenancy architecture with site subscriptions
Claims-based authentication
Profile synchronization
High availability automatic failover
Content type syndication
Sandbox solutions
Policy
Flexible deployment
Prerequisite installer
Visual Upgrade
Configuration wizards
Scriptable deployment
On-premise or online

Before we begin the discussion of each of the features, let's briefly mention how these features can be used to provide governance, a very key topic for IT pros.

Governance

Governance in Microsoft® SharePoint® Server 2010 is the set of roles, responsibilities, and processes that you put in place in an enterprise to guide the development and use of a solution based on SharePoint Server. There are several new features in SharePoint Server that help the IT pro deliver governance: metadata creation and management, shared service applications, taxonomy, multi-tenancy, sandbox solutions, content type syndication, and policy. These topics are discussed in greater detail in the guide but it is worth mentioning them now because it is important to realize how features in SharePoint Server can provide solutions. Solutions generally utilize several features together, and it is easy to overlook the solution capability when discussing features individually. Another very important solution is high availability. This is discussed in the next section.

High Availability

Several feature improvements targeted at the database level and the search service help achieve high-availability architecture. Microsoft® SharePoint® Server 2010 leverages the underlying Microsoft SQL Server™ high- availability technologies such as log shipping, snapshots, and database mirroring, and now includes automatic failover. Other features such as read-only content databases, unattached database backup and restore, and configuration database backup and restore provide much more granular control and manageability of the SharePoint farm's data. Now it's time to discuss some of features beginning with SharePoint deployment and upgrading from Office SharePoint Server 2007.

Deployment and Upgrade

System Requirements

As one would imagine, a powerful application such as this requires a server environment capable of supporting it. As such, the requirements for your infrastructure will draw upon the latest server and application technologies. To implement a Microsoft® SharePoint® Server 2010 environment, your infrastructure must leverage the following minimum requirements:

- Microsoft Windows Server® 2008 64-bit operating system with Service Pack 2 (SP2) or later OR Windows Server 2008 R2
- Microsoft SQL Server™ 2005 64-bit with SP2 or later or SQL Server 2008 64-bit
- Microsoft .NET Framework 3.5 with SP1 installed

Level 1 Internet browser options running on the Windows® operating system:

- Windows Internet Explorer® 7 32-bit
- Internet Explorer 8 32-bit
- Firefox 3.x 32-bit

Level 2 Internet browsers options:

- Internet Explorer 7 64-bit
- Internet Explorer 8 64-bit
- Firefox 3.x on non Windows operating system
- Safari 3.x

For the complete and updated list of prerequisites for installing SharePoint Server 2010, see "Determine hardware and software requirements (SharePoint Server 2010)" (http://go.microsoft.com/fwlink/?LinkId=166546) on the Microsoft TechNet Web site.

Installation

For administrators who have previously installed SharePoint Server, the installation options for SharePoint Server 2010 will be familiar. In fact, a number of enhancements help make the installation process easy for all users. To start, Microsoft has added the prerequisite installer. Before installing SharePoint Server 2010, run the prerequisite installer and it will scan the system to ensure that all of the components required to deploy SharePoint (Internet Information Services, .NET Framework, and so on) have been installed. If they haven't, the prerequisite installer will download and install the components needed for the SharePoint Server 2010 installation. Optionally, you can specify that the installer pull files from a local path to avoid the trip to the Internet.

After the prerequisite components have been installed, you have two options for installing SharePoint Server 2010: use the graphical user interface (GUI) or script the installation by using configuration files and Windows PowerShell™. The GUI provides a simple step-by-step process where users can specify the configuration they want. Using Windows PowerShell is a bit more involved, but in the end it usually proves to be the

better choice. By scripting the installation, you can ensure that all of your servers running SharePoint Server have identical configurations. Scripted installations are also advantageous from a disaster recovery perspective. If a server goes down, you can reinstall SharePoint Server quickly by using a previously created script. Both methods are fully supported.

Another new aspect of the installation process is the addition of a farm passphrase. This is a password that is used to encrypt all communications across the farm. The farm passphrase is required before you can remove a server from the SharePoint farm or add a server to the farm. The passphrase can be changed through Windows PowerShell by the administrator.

Other changes allow network administrators to organize and manage servers running SharePoint Server on the network. Active Directory® markers make it possible to add new servers running SharePoint Server to an organizational unit (OU) in Active Directory when they are installed on the network. This makes it easier for network administrators to keep track of all servers running SharePoint Server within their domain. Administrators can use Group Policy to manage farm proliferation on the network, for example to prevent SharePoint from being installed on unauthorized servers. This gives administrators tighter control over the SharePoint environment and increases their ability to enforce governance policies.

Upgrade from Office SharePoint Server 2007

So many people have invested so much in Office SharePoint Server 2007 — the same level of investment has been made in upgrade processes and strategies for moving to SharePoint Server 2010. These investments were made in all phases, from preparing to upgrade to customizing the post-upgrade environment.

Preparing for Upgrade

The first investment was released in Office SharePoint Server 2007 Service Pack 2, with the addition of the Stsadm.exe operation, **preupgradecheck**. This operation can be run on your Office SharePoint Server 2007 farm to report on farm and server data. It will identify key information, such as:

- Servers and total amount of content
- Search configuration
- Alternate access mappings

- Features
- Site definitions
- Language packs

It will also expose potential issues, including:

- Large lists
- Data orphans
- CAML views and CAML content types
- Modified content databases that have improper schemas
- Missing upgrade dependencies

All of this information will be collected and then output into an .htm file you can view from your browser. This process is read-only. It does not change your environment, so you can run it often, not only before upgrading but also just to check the general health of your environment. Any automatic fixing of your environment will occur at upgrade time and not during pre-upgrade scanning.

If you are already running SharePoint Server 2010 and would like to check whether your content databases are Office SharePoint Server 2007 or SharePoint Server 2010 databases, you can use a Windows PowerShell cmdlet. The cmdlet **Test-SPContentDatabase** can be run against the databases of either version. It will test a specific database referenced against a specific Web application. It identifies current or potential issues, such as:

- Data orphans
- Missing site definitions
- Missing features
- Missing assemblies

This cmdlet is meant to complement the pre-upgrade checker report and is also read-only, preventing any disruption of your data while you are testing for issues.

Upgrade Methods

There are two primary methods for upgrading from Office SharePoint Server 2007 to SharePoint Server 2010: in-place and database attach. The in-place method is used to upgrade an existing Office SharePoint Server 2007 server to SharePoint Server 2010, and the process can be restarted as necessary if issues arise. While the database attach

method would allow you to backup a Office SharePoint Server 2007 database and attach it to your SharePoint Server 2010 Web application, SharePoint Server would then upgrade this database and make it available through the Web application. Additionally there is support for single click installations to be upgraded by using in-place upgrade, and then migrated from a Windows Internal Database deployment to take advantage of remote BLOB storage (RBS, covered later in this guide).

Downtime Mitigation

Several features have been introduced in SharePoint Server 2010 to allow for upgrade to occur with little to no outages and at the same time to allow for quicker upgrade windows.

The first feature is the use of read-only databases, which was made available starting with Office SharePoint Server 2007 Service Pack 2. SharePoint Server 2010 now recognizes a content database has been made read-only from within SQL Server and will trim the user interface as if all site collections were marked read-only. This allows users read access to the content while the upgrade takes place.

Another advancement to reduce the amount of time an upgrade will take is that SharePoint Server 2010 supports performing several database attach upgrades at the same time. Through the use of multiple Windows PowerShell sessions, multiple databases are upgraded. which means the amount of data upgraded at one time is limited only by your SQL Server resources.

Finally, for customers who have such extreme amounts of content that completing the upgrade within a reasonable window is not possible, there is another option. This option involves using alternate access mapping redirections to direct traffic between a SharePoint Server 2010 farm and Office SharePoint Server 2007 farm based on the requested URL by using client-side 302 redirects.

Visual Upgrade

To mitigate the impact of the upgrade on the user community, SharePoint Server 2010 supports Office SharePoint Server 2007 master pages and cascading style sheets. By default, after a content database is upgraded, the sites will be displayed with the Office SharePoint Server 2007 visuals, giving the user their familiar look and feel. An upgraded site can then exist in one of three states: Office SharePoint Server 2007, SharePoint Server 2010 preview mode, and SharePoint Server 2010. This allows the site

administrator to preview how the site looks with the SharePoint Server 2010 user interface before committing to it. This setting is at the site level, allowing for a very granular, flexible experience.

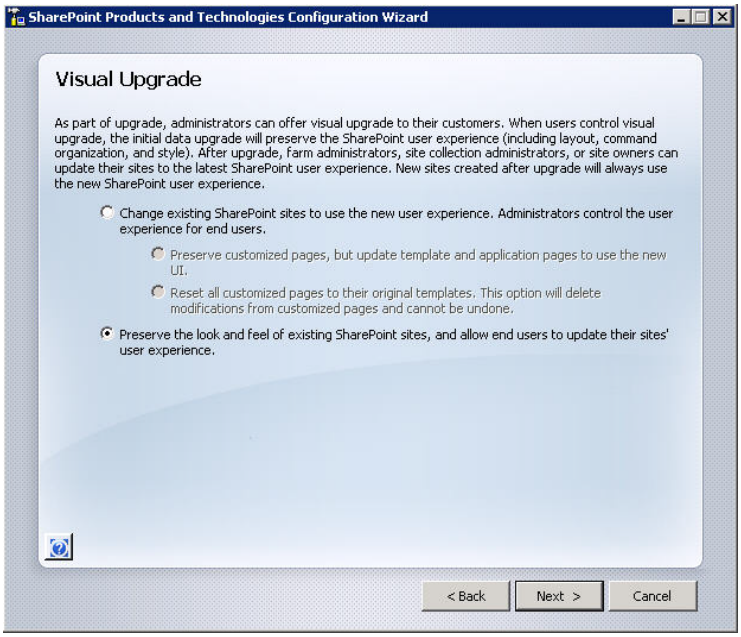
SharePoint Products and Technologies Configuration Wizard
Visual Upgrade
As part of upgrade, administrators can offer visual upgrade to their customers. When users control visual upgrade, the initial data upgrade will preserve the SharePoint user experience (including layout, command organization, and style). After upgrade, farm administrators, site collection administrators, or site owners can update their sites to the latest SharePoint user experience. New sites created after upgrade will always use the new SharePoint user experience.
Change existing SharePoint sites to use the new user experience. Administrators control the user experience for end users.
Preserve customized pages, but update template and application pages to use the new UI.
Reset all customized pages to their original templates. This option will delete modifications from customized pages and cannot be undone.
Preserve the look and feel of existing SharePoint sites, and allow end users to update their sites' user experience.
< Back
Next >
Cancel

Figure 1: Visual Upgrade Options

Upgrade Logging

The logging capabilities have been expanded and standardized, allowing for easier, more consistent reporting on the upgrade process. This includes the creation of a unique log for each upgrade. Also, an errors-only log is generated, which reduces the need to look through the full logs to discover issues.

Upgrade Scenario

To better understand the upgrade process, consider the following scenario. Randy has been tasked with upgrading his Office SharePoint Server 2007 farm to SharePoint Server 2010 while incurring as little downtime as possible. It will be acceptable for the data to be read-only during the upgrade. Randy will be deploying SharePoint Server 2010 on new hardware and will be performing the upgrade by using the database attach method.

To begin the process, Randy has tested and updated the server running Office SharePoint Server 2007 with Service Pack 2 and all applicable cumulative updates. Now he can run **Stsadm.exe –o preupgradecheck**. He uses the output to verify that his farm

is ready to be upgraded. He does discover some issues where he is not following SharePoint best practices around large lists, and is able to examine the list and find solutions. Because **preupgradecheck** is a read-only tool, it is safe for him to run it often, confirming his progress as he works to mitigate the issues that have been raised.

Now that Randy has his databases in good shape, he begins building SharePoint Server 2010 on his new hardware. After he has established this farm, configured all necessary service applications, and provisioned his required Web applications, he can begin the process of testing upgrade. Because he has chosen the database attach method, he performs a backup of his database from production and restores it to his new computer running SQL Server. On this SQL Server, he takes advantage of the Windows PowerShell cmdlet **Test-SPContentDatabase** to report any potential issues with attaching the content database to his newly provisioned Web application. From the output of this command he finds he has not installed all of the necessary Features on his new server. He is now able to install the necessary Features avoiding a failed upgrade.

With all of the necessary pieces now in place, Randy performs his first upgrade by using the Windows PowerShell cmdlet **Upgrade-SPContentDatabase**. After the upgrade concludes, he reviews the errors-only upgrade log and then the full upgrade log, which gives him both a focused picture of any issues that occurred and the line-by-line story that he can review to get a better understanding of what happened during the upgrade process.

With the upgrade cmdlet successfully completed , he now can open the site and review his content. The first thing he notices is that his site looks exactly as it did in the Office SharePoint Server 2007 farm, thanks to the Visual Upgrade features. To see what his site will look like with the SharePoint Server 2010 look and feel, he goes into Site Settings and selects the preview option. This gives Randy the opportunity to confirm that everything looks appropriate with the new visuals applied. If he is happy with the results, he can make the change permanent; if he feels he will need to make adjustments to the site, he can turn preview off and return to the Office SharePoint Server 2007 interface. The feature is set at the individual site level, allowing Randy to make very granular choices. He can also use Windows PowerShell to script the changing of the visuals.

After Randy finishes working with the various business content owners to validate the upgrade process and works through any issues, he is ready to plan his production upgrade schedule. His plan is to notify his users and then take his Office SharePoint

Server 2007 farm and set all of the content database to read-only from within SQL Server. Not only will this set the data to read-only, preventing any data inconsistency during the upgrade window, it will trim the user interface by taking away New or Edit functionality for all users, which helps avoid confusion. After all databases are read-only, Randy will perform a backup of the databases and restore them to his new computer running SQL Server. With the databases in-place he can then begin attaching the databases by using Windows PowerShell. During testing, he confirmed his hardware was powerful enough that he could open multiple Windows PowerShell windows, each one upgrading a different database, all at the same time. This allowed him to shorten his upgrade window.

With all of his company's SharePoint content upgraded and available, Randy then had DNS updated to resolve all of his Web applications to the new farm. His content owners then confirmed the production upgrade was successful and switched their visuals to SharePoint Server 2010 as appropriate. Each content owner had previously determined the necessary course of action for moving to the new visuals and the time frame for making the change. Randy has set a hard date of 30 days before he will use Windows PowerShell to force all visuals to the SharePoint Server 2010 look and feel.

Randy has had a very successful upgrade to SharePoint Server 2010 thanks to his using out-of-the-box SharePoint tools and performing lots of planning and testing. Because he worked out the kinks throughout the testing phase, he was able to accomplish the upgrade with minimal impact on his business users.

Patch Management Improvements

While upgrade is used to move from version to version, updating (also called "patching") uses the same principles to move from build to build. To allow control and flexibility of the update process, several capabilities are available. The update management user interface and update reporting cmdlet allow for the monitoring of updates throughout the farm. There are also update status health rules that will alert an administrator to inconsistencies. To allow for more flexibility while applying updates in farm scenarios, SharePoint Server 2010 supports backward compatibility of updates. This will permit an administrator to apply updates to the servers in the farm without applying the updates to the databases for a temporary period. This allows the scheduling of downtime for the database upgrade portion of the update, or the use of the downtime

mitigation technique of using a read-only database to avoid an outage while rolling in updates.

SharePoint Central Administration Web Site

The Central Administration Web site has been redesigned in Microsoft® SharePoint® Server 2010 to provide a more familiar experience and make it easier for users to find what they are looking for. The home page for Central Administration groups major functional areas together and lists many of the most common tasks. Each of the major areas (Application Management, Monitoring, and so on) is represented on the Home page and can be accessed by clicking its name or by clicking the corresponding link from the navigation at the left side of the page.

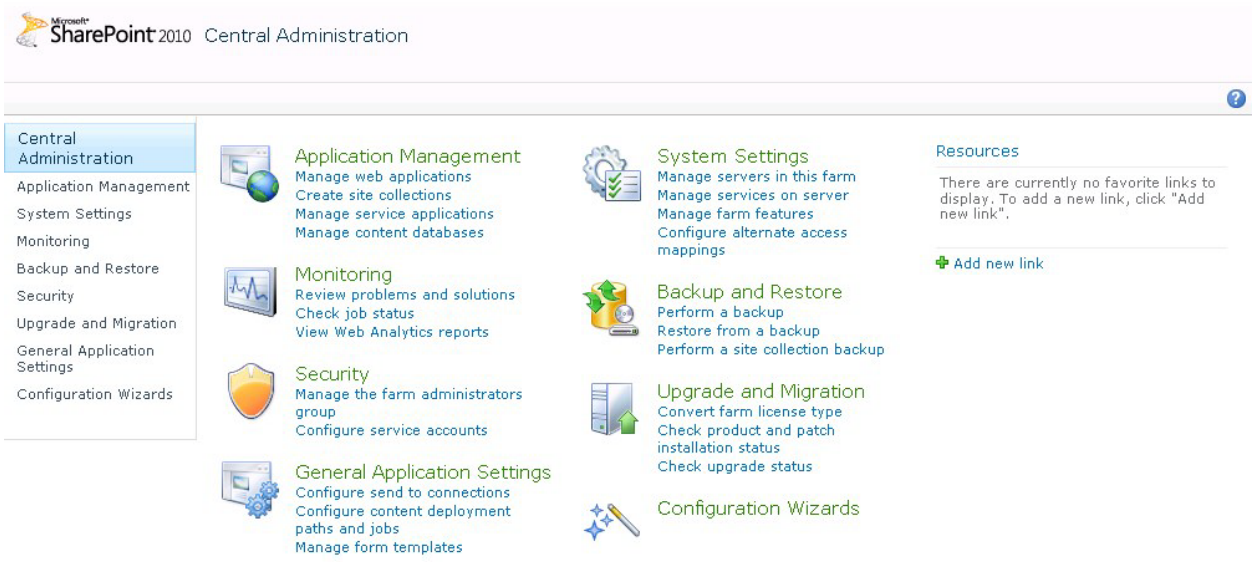

Figure 2: Central Administration

Although the user interface has changed slightly, users will find many of the menus to be familiar, with some new available options and functionality. One of the places where this is more obvious has to do with how Web applications are managed. Web applications are still created by following the same process, but the process is now managed through a ribbon interface on the Manage Web Applications page. The ribbon makes it easier for users to view or change details about a Web application by supplying all the options so they are one click away. In Office SharePoint Server 2007, many of these tasks required the user to click a different menu and reselect the Web application

each time. Now, users simply select the Web application and then all of the options for managing the Web application can be accessed by a single click.

Figure 3: Ribbon Interface

In addition to the more obvious changes to the user interface, there are several new pieces of functionality hiding among the menus. As you can see in the screenshot of the Central Administration home page, a new page named Configuration Wizards has been added. Wizards are designed make the configuration process easier by guiding users through the steps to configure functionality.

Figure 4: Configuration Wizards page

The Farm Configuration Wizard is available by default. This wizard takes users through the process of specifying the accounts that each service will run under. It is possible for additional wizards to be added by third-party vendors or by developers to assist with configuring other areas of SharePoint Server 2010.

Clicking the Backup and Restore link displays another new set of features. Previously the ability to perform granular backups such as backing up a site collection, site, or list was only possible by using the Stsadm command line tool. New in SharePoint Server 2010 is the ability to perform these granular backups directly from the Backup and Restore page in Central Administration.

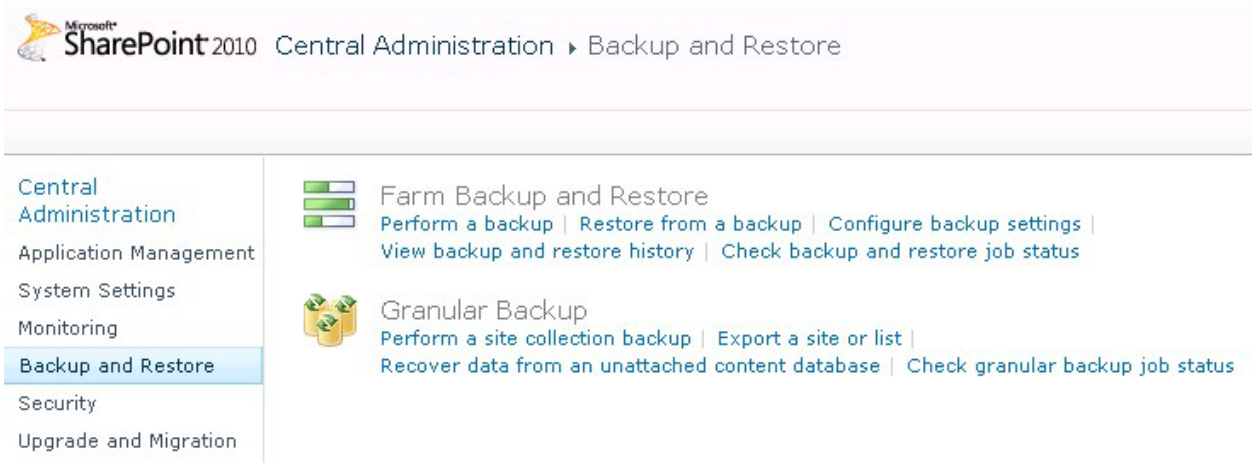
SharePoint 2010
Central Administration ▸ Backup and Restore
Central Administration
Application Management
System Settings
Monitoring
Backup and Restore
Security
Upgrade and Migration
Farm Backup and Restore
Perform a backup | Restore from a backup | Configure backup settings |
View backup and restore history | Check backup and restore job status
Granular Backup
Perform a site collection backup | Export a site or list |
Recover data from an unattached content database | Check granular backup job status

Figure 5: Backup and Restore Page

A complete disaster recovery plan not only includes the ability to restore servers and databases but also how to recover smaller units of data like a single document. In the past, this scenario proved problematic — recovering a single file often required an administrator to restore an entire content database to a separate SharePoint farm. Moving whole databases around and restoring them to new farms was a lot of work to recover a small amount of data. SharePoint Server 2010 has added the ability to recover data from an unattached content database. In other words, if you need to restore data from a backed-up content database, you can browse the content of that content database as long as it is attached to a computer running SQL Server, even if it is not necessarily associated with SharePoint Server. Administrators can browse the content database, back up a site collection, or export a site or list. After the content has been located and saved into a separate location, it can quickly be restored to the production database. The new backup and restore functionality will be covered in more detail later in this guide.

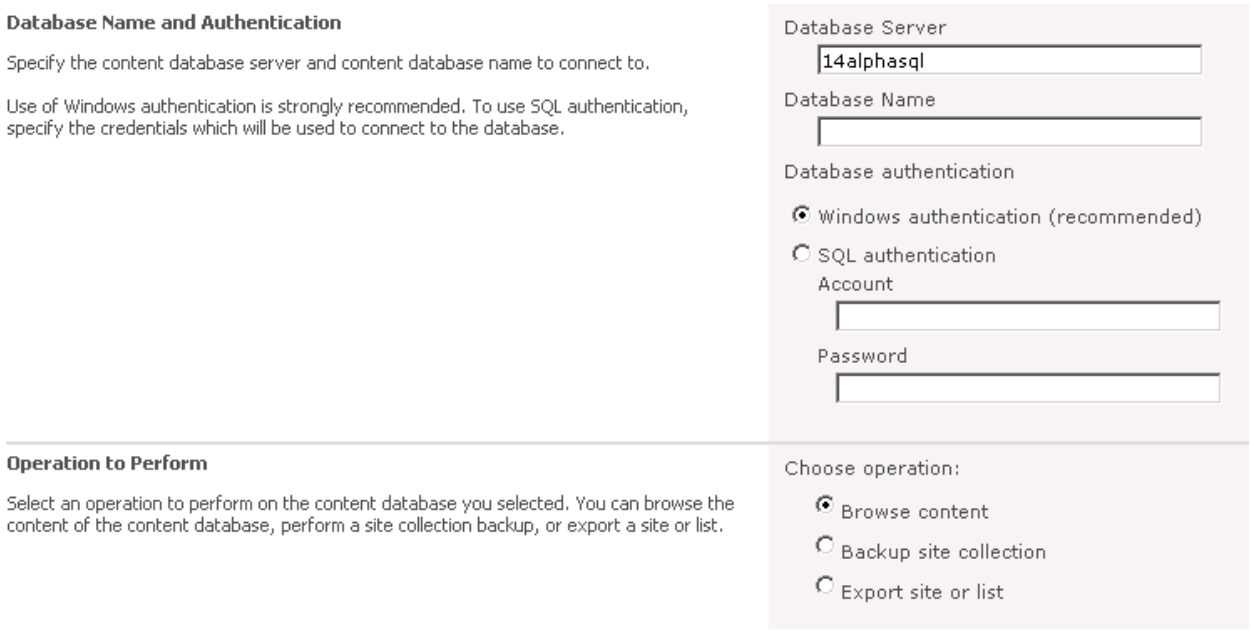
Database Name and Authentication
Specify the content database server and content database name to connect to.
Use of Windows authentication is strongly recommended. To use SQL authentication, specify the credentials which will be used to connect to the database.
Database Server
14alphasql
Database Name
Database authentication
Windows authentication (recommended)
SQL authentication
Account
Password
Operation to Perform
Select an operation to perform on the content database you selected. You can browse the content of the content database, perform a site collection backup, or export a site or list.
Choose operation:
Browse content
Backup site collection
Export site or list

Figure 6: Connect to a content database

The security model in SharePoint Server 2010 has been expanded to allow administrators more control over the management of service applications. In Office SharePoint Server 2007, farm administrators had the ability to manage services on the server. Farm administrators can now click a service application and be presented with several options for managing the service, including specifying administrators for the service and setting permissions. Service applications will be covered in more detail later in this guide.

Service Applications

Microsoft® SharePoint® Server 2010 implements a new, more flexible shared service model compared to its predecessor, Office SharePoint Server 2007. This new model is referred to as service applications. The service application architecture is now also built into Microsoft SharePoint Foundation 2010, in contrast to the Shared Services Provider (SSP) architecture that was only part of Office SharePoint Server 2007. A key benefit here is that all services are installed by default and there is no SSP setup. The figure below shows the Applications Management page in Central Administration, illustrating the Service Applications option.

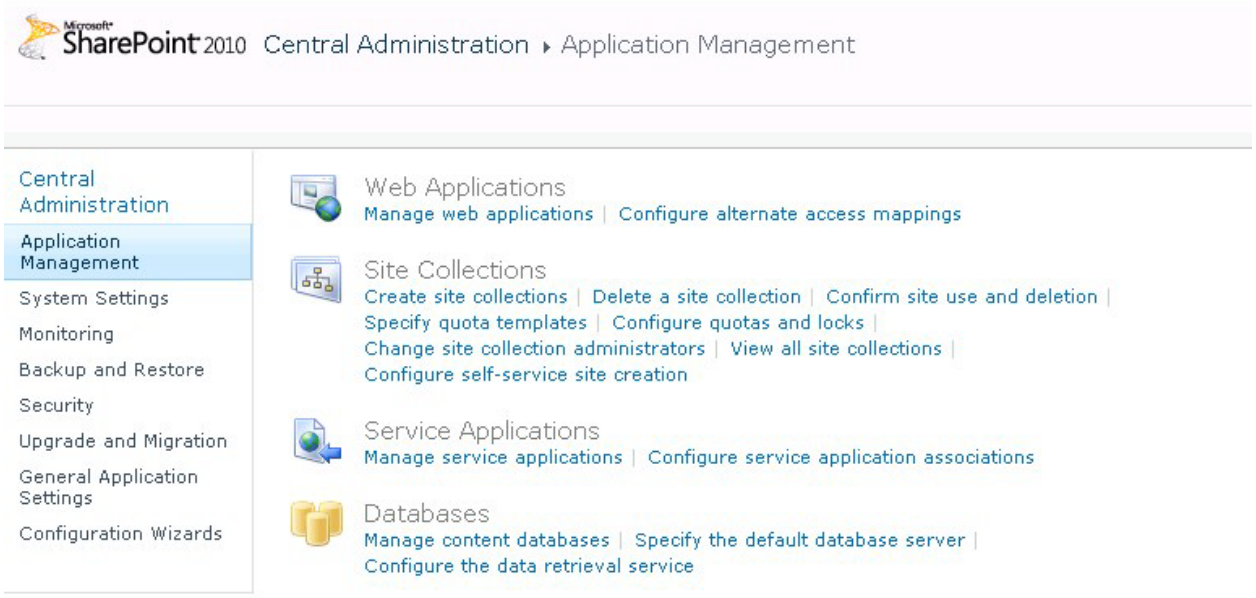
SharePoint 2010 Central Administration ▸ Application Management
Central Administration
Application Management
System Settings
Monitoring
Backup and Restore
Security
Upgrade and Migration
General Application Settings
Configuration Wizards
Web Applications
Manage web applications | Configure alternate access mappings
Site Collections
Create site collections | Delete a site collection | Confirm site use and deletion |
Specify quota templates | Configure quotas and locks |
Change site collection administrators | View all site collections |
Configure self-service site creation
Service Applications
Manage service applications | Configure service application associations
Databases
Manage content databases | Specify the default database server |
Configure the data retrieval service

Figure 8: Application Management

The key limitation of the SSP architecture was that it was configured by using a set of services, and all Web applications associated with the SSP bore the overhead of all the services even if they weren't being used. To change the service configuration for a particular Web application, a new SSP would have to be created. The service application architecture allows a set of services to be associated with a given Web application and a different set of services to be associated with another Web application. Also, the same service application can be configured differently in different Web applications; therefore, Web sites can be configured to use only the services that are needed, rather than the entire bank of services. Similar to the SSP model in Office SharePoint Server 2007, a single set of services can be shared by all sites in a farm if this is a requirement. The figure below shows the Manage Service Applications page ribbon.

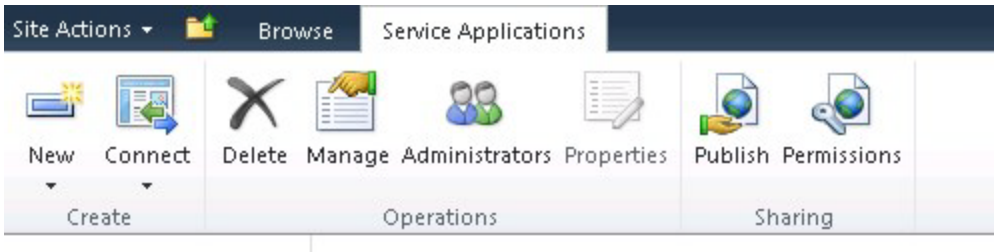

Figure 9: Service Application Management user interface

By publishing a service application, you can share it across server farms. This capability does not apply to all service applications, and some services can be shared only within a single server farm. Services that support sharing across farms can be run in a central farm and consumed from regional locations. From the ribbon, you can also see that a service application can connect to other service applications published by other server farms. This sharing capability allows some services to be hosted locally while sharing those of other farms.

Additional improvements for the service application model include:

- The services architecture is extensible, allowing third-party companies to build and add services to the platform.
- Services are managed directly in Central Administration (rather than a separate administration site).
- Services can be monitored and managed remotely.
- Services can be managed and scripted by Windows PowerShell™.
- Shared services communications take place over HTTP(S). Shared services do not directly access databases across farms.
- Most new services are built on the Windows® Communications Framework. They have optimization built into their protocol, using binary streams instead of XML for data transfer. Test results show improvements in network throughput with this change.

Examples of service applications are described in the following sections.

Search

SharePoint Server 2010 Search includes new features and a new architecture that enables a more scalable topology. Search capability continues to be pervasive and integrates very well with the new social networking features. The new architecture scales to approximately 100 million items, and search can be used in a multi-tenant hosting environment.

Search management has been improved by consolidating search administration to a single dashboard in the user interface, and administration can be scripted by using Windows PowerShell. Password management can be achieved by using managed accounts. Search performance and functionality can be monitored, and there is also support for System Center Operations Manager (SCOM) monitoring and alerting. Search reporting has been improved by using built-in and extensible search analytics and reporting engine.

Search Service

Search uses the new services architecture described earlier. This new architecture allows farms to connect to multiple farms to consume cross-farm services. In large environments, an entire enterprise services farm — which is a farm that hosts the most commonly used cross-farm services, including search — can be deployed. A dedicated search farm, which is a farm that is optimized to provide search, can also be implemented.

Search Architecture

Search in SharePoint Server 2010 has been re-architected to allow greater redundancy within a single farm and improvements to scaling up and out. The query architecture and the crawling architecture can be scaled out separately, based on the needs of an organization, thus providing greater flexibility.

Query Architecture

The query architecture includes query servers, index partitions (which reside on query servers), and property databases. An index partition represents a portion of the entire index, and therefore the index is the aggregation of all index partitions. Partitioning the index allows different portions of the index to be spread across query servers. Administrators decide on the number and configuration of each of the partitions. At least one server in a farm must host the query role, and more query servers can be added to increase performance. Two or more query servers provide redundancy based on the configuration of index partitions. For example, a farm with three query servers can be configured so that each query server has an index partition that represents one-third of the index. Redundancy for the query servers can be achieved by creating a second instance of each index partition on another query server. Deploying index partitions across query servers can help balance the query-processing load, provide redundancy, and increase query performance.

The query server receives a query and forwards the request to all query servers to process (across all index partitions). The query server then merges the results to display to users.

Crawling Architecture

The crawl server hosts the crawling architecture, which includes crawlers, crawl databases, and property databases. The search architecture can be scaled out based on crawl volume and performance requirements. At least one crawler component is

present, and it is the responsibility of each crawler to crawl content. Each crawler is associated with a crawl database, and the crawled content and history are stored in the crawl database. Multiple crawlers can be used to crawl different content simultaneously. This improves performance and can also provide redundancy. Crawlers reside on crawl servers, populate index partitions, and propagate the partitions to query servers. Property information is stored in the property database. The number of property databases depends on the volume of content that is crawled and the amount of metadata that is associated with the content.

The index role must be hosted on at least one server in the farm. Two or more crawl servers provide redundancy based on how crawlers are associated with crawl databases. Additional crawl servers can be added to increase performance and to scale for capacity. Below is an example of a scaled out search architecture.

Combined Web and query servers
Place the Web server role on the same servers with the query role.

There are two index partitions. For redundancy, a copy of each index partition (a index partition instance) is on each computer.

Index servers
Administration component is on one server.

Each server has one crawler. Both crawlers are associated with the crawl database.

Database servers (clusters or mirrors)
One database server set for:

- Search admin database.
- One property database.
- All other SharePoint databases.

One database server set containing the crawl database, which is isolated for performance reasons.

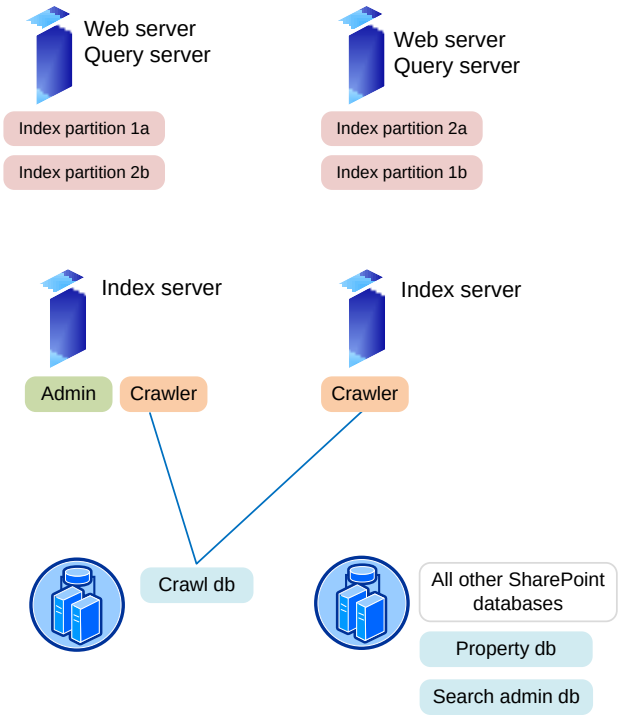

Figure 10: Search Architecture

Improving Search Performance

Many times the administrator is tasked with improving different aspects of the overall search process. The following table highlights some options available to the administrator based on the new search architecture.

Aspect to improve	Action
Full crawl time and result freshness	Add crawl servers, crawlers, and crawl databases. Each crawl database can contain content from independent sources. Each crawl database can have several crawlers associated with it, and those crawlers can be distributed among multiple crawl servers. Multiple crawlers and associated crawl databases allow you to crawl multiple content repositories.
Query response time	Add query servers and index partitions to improve query latency caused by high peak query load. Isolate the property database from crawl databases by moving it to a separate database server if high query latency is caused by database load. Add index partitions. Each index partition can contain up to ~10 million items, and can be cloned by deploying multiple instances across several query servers. Query throughput increases when you add query partition instances.
Query functionality availability	Deploy redundant query servers, redundant query partitions and partition instances. Use clustered or mirrored database servers to host crawl and property databases.
Content crawling and indexing functionality availability	Use multiple crawlers per crawl server, and use redundant crawl servers. Add crawl databases. Crawlers associated with a given crawl database can be distributed across crawl servers for availability and load distribution.

FAST Search for SharePoint

FAST Search for SharePoint is a new Microsoft product that is integrated with SharePoint Server 2010 to provide high-end search capability. It is highly efficient and can search a huge corpus of data extremely fast. It features an outstanding search user experience out of the box, granular control of user experience and relevance, powerful content processing capabilities, and platform flexibility and scaling to handle the most challenging search applications. FAST Search for SharePoint is not part of the Technical

Preview; it will be available with the public Beta. The FAST Search Site template is included with the Technical Preview. However, it will not function without FAST Search servers, so it should not be used with the Technical Preview.

Business Connectivity Services

Microsoft Business Connectivity Services (BCS) provide the capability to connect SharePoint Server 2010 and Office 2010 client applications to external data sources (such as SQL, Oracle, SAP Web services, or custom applications). BCS can also be called the "new BDC," because is replaces the Business Data Catalog, introduced in Office SharePoint Server 2007, but it includes much more capability. The BCS is actually a platform that provides a SharePoint-based framework for creating composite applications, which are built by combining services and functionality from other enterprise applications to address the collaborative needs of users. These composite applications use the Office client user interface in addition to the functionality of the application. This helps people perform their tasks within the familiar user interface of the Office and SharePoint Server systems, and allows people to take actions and make decisions within the context of the problem at hand.

The configuration and administration of the BCS is primarily accomplished by using the BCS application, but some configuration is also done by using the Secure Store Service application. The BCS application is a shared service that takes advantage of the shared services architecture discussed above. As mentioned, multiple BCS applications can be configured independently by different sets of administrators and multiple instances of the same service can run on the same farm, each having a unique set of administrators. A specific application can be shared across server farms, so that an instance can be run in a central farm and consumed from regional locations so that a composite application could be made available across these locales. The first step in using the BCS is to create a new service application and model.

Business Data Connectivity Service Application and Model

A new Business Data Connectivity Service application is created by using one of the following methods:

- Selecting services while running the Initial Configuration Wizard and choosing the Business Connectivity Service.

- Adding the Business Data Connectivity service application by using the New button on the Manage Service Applications Web page ribbon in the Central Administration site as illustrated earlier.
- Using Windows PowerShell.

After the application has been created, the administrator will associate the application with a database and a service account. The application can also be partitioned for multi-tenancy if necessary. The external data source instance specifies particular connection and authentication information for an external data system. Essentially, an application model describes an external data source. The model is a collection of metadata that provides a framework for converting operations requested by SharePoint Server or a client application into requests that are specific to a given external data source. More specifically, the application model contains the XML descriptions of one or more external content types.

External Content Types

An external content type represents a reusable description of an object that can be used in a composite application, such as "Customer," "Order," or "Contact" (recall that these were formerly called BDC entities). The external content type also includes the object's data fields; the methods to create, read, update, or delete that object; actions that users can take on the object; and information that supports connecting to the external data source where the object is stored. Notice that both read and write capability is included, which is an improvement over the BDC capability in Office SharePoint Server 2007. Also, batch and bulk operation support is available. External content types represent data that is stored in an external data source, such as Microsoft SQL Server™ and other relational databases, SharePoint Server, and Web services. External content types are created by using SharePoint Designer 2010, Visual Studio® 2010, and by importing an application model that contains one or more external content types into a BCS service application.

SharePoint Server and Office Integration

BCS data can be accessed by using a Web browser and displayed inside SharePoint sites by using Web Parts and external lists. Business data can be exposed as Office Word document properties and inserted into documents. Users can interact with business data by using the familiar Microsoft Office Outlook® interface, and take the data offline by using the new SharePoint Workspace (the new Groove application). External content types can also be consumed natively by Microsoft InfoPath® 2010, Microsoft Access™

2010, and other Office applications via custom code. SharePoint search can also use the BCS to index external data; therefore, Office applications and SharePoint Server use the BCS to access external data by using external content types.

Managed Metadata Service

The managed metadata service is a service application that enables the use of managed metadata and allows you to share content types across site collections and across Web applications. The service publishes a term store and content types, and the metadata is consumed via the managed metadata connection. A managed metadata service and connection are created when the metadata service application is created.

Managed Metadata Service Application

Creating the service application requires the administrator to specify the database to be used as the term store. When you create new managed terms, or when users add managed keywords, these terms are stored in the database. Like other service applications, the managed metadata service can be published to provide access to other Web applications. When a service application is published, a URL to the service is created. The administrator of another Web application can create a connection to your service by using this URL.

In addition to sharing metadata, you can also use the managed metadata service to share content types. By creating a new managed metadata service and specifying a site collection as the content type hub, you can share all content types in the site collection's content type gallery.

You can create multiple managed metadata service applications. This provides the capability to share multiple term stores and content types from multiple site collections. Each service must specify a different term store during the creation process, and a new database will be created if it does not exist.

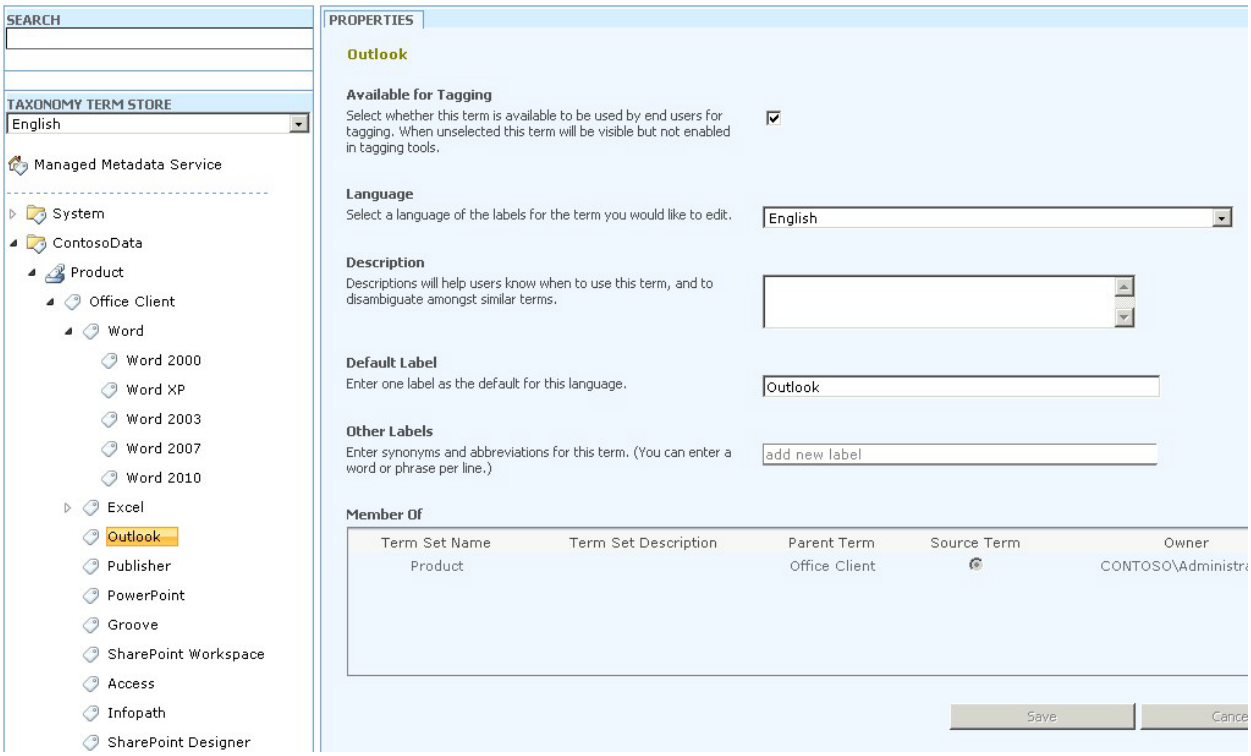

Figure 11: Manage Terms

Managed Metadata Service Connection

The connection provides access to the service. When you create a managed metadata service, a connection to the service is automatically created in the same Web application as the service. Web applications can have connections to multiple services. Several properties are specified as part of the connection. Values for each of these properties must be specified as part of configuring the service connection.

Default keyword location Specifies that the term store will be used to store new managed keywords. This provides the capability for sharing and managing a common set of keywords across site collections.

Default term set location Specifies where to store term sets when new site columns are created. This provides the capability for sharing and managing a common set of terms across site collections.

Use content types Makes content types associated with this managed metadata service available to users of sites in this Web application.

Push-down content type publishing updates from the content type gallery to subsites and lists by using the content type Whether to update existing instances of the changed content types in subsites and libraries.

Example Scenario

Consider a company that has four departments: IT, HR, Products, and Legal. IT, HR, and Legal each have their own site collection that serves as each department's portal, while there are several product team site collections, one for each product team. All these site collections are contained in the same Web application. Two governance requirements have been specified for managing metadata: All documents that are created must include a core set of properties, and all keywords must be stored centrally.

To meet the first requirement, IT creates a content type called Document-Base at the root site collection of its department portal. IT adds columns to Document-Base for all of the required properties. Then IT creates a managed metadata service application and specifies the root site collection as the hub of the content types it is sharing. IT publishes the service and provides the service's URL to all departments. A connection to IT's managed metadata service is created. The second requirement — that all keywords be stored centrally — can be satisfied when each site collection connects to the service hub by specifying that managed metadata service as the default keyword store.

This scenario illustrates how the metadata service can be configured so that all departments have access to a centrally managed set of metadata defined in the Document-Base content type, and all keywords can be centrally managed and stored in a single location (IT's term store), so all departments can use all of the keywords.

Claims-based Authentication

SharePoint Server 2010 incorporates a new, more powerful and flexible authentication model that works with any corporate identity system, including Active Directory® directory services, LDAP-based directories, application-specific databases, and new user-centric identity models such as LiveID. This model uses claims-based authentication and a new product, code-named "Geneva." Claims-based authentication is built around the concept of an identity and is based on standards — WS-Federation, WS-Trust — and protocols like the Security Assertion Markup Language (SAML). An identity is a set of information about a user, such as name, e-mail address, department, and so on. "Geneva" is actually three related technologies: Active Directory Federation Services (formerly known as Geneva Server), Windows Cardspace™ (formerly known as

Cardspace Geneva), and Windows Identity Foundation (formerly known as the Geneva Framework). Why use this new system? Identity drives very important aspects of an application such as identifying who the user is (authentication), what permissions the user is granted (authorization), and how the application interacts with the user (personalization). All applications work with identity today, but usually in their own way. Often, integrating these applications can be a very difficult process.

Claims-based identity provides a common way for applications to acquire identity information from users inside their organization, in other organizations, and on the Internet. Identity information is contained in a security token, often simply called a token. A token contains one or more claims about the user. Think of it as metadata about the user that stays with them throughout their session.

Claims-based authentication opens the door to great possibilities in SharePoint Server 2010. Greater flexibility for authenticating users against multiple repositories within the same Web application and a more personalized experience based on the claims or information about the user provided in the token.

Service Applications Conclusion

Service applications are a modular set of services that will provide SharePoint administrators with needed flexibility to provide only necessary functionality to Web applications within the farm or the enterprise. They can be scaled out to achieve maximum performance or combined into groups to maximize resources. Because each one can be assigned and administered separately, SharePoint administrators will be able to delegate as appropriate.

Health and Monitoring

Microsoft® SharePoint® Server 2010 includes a number of features that provide the administrator with tools for monitoring the health and performance of the SharePoint farm. These features are categorized into the following groups: diagnostics, reliability and monitoring, and reporting. The following table shows the various categories and their corresponding features.

Category	Functionality
Diagnostics	Unified Logging Service (ULS) Usage database Developer dashboard
Reliability and Monitoring	SharePoint Maintenance Engine (SPME) Rules System Center Operations Manager (SCOM) monitoring
Reporting	Out-of-box usage reports SCOM reports

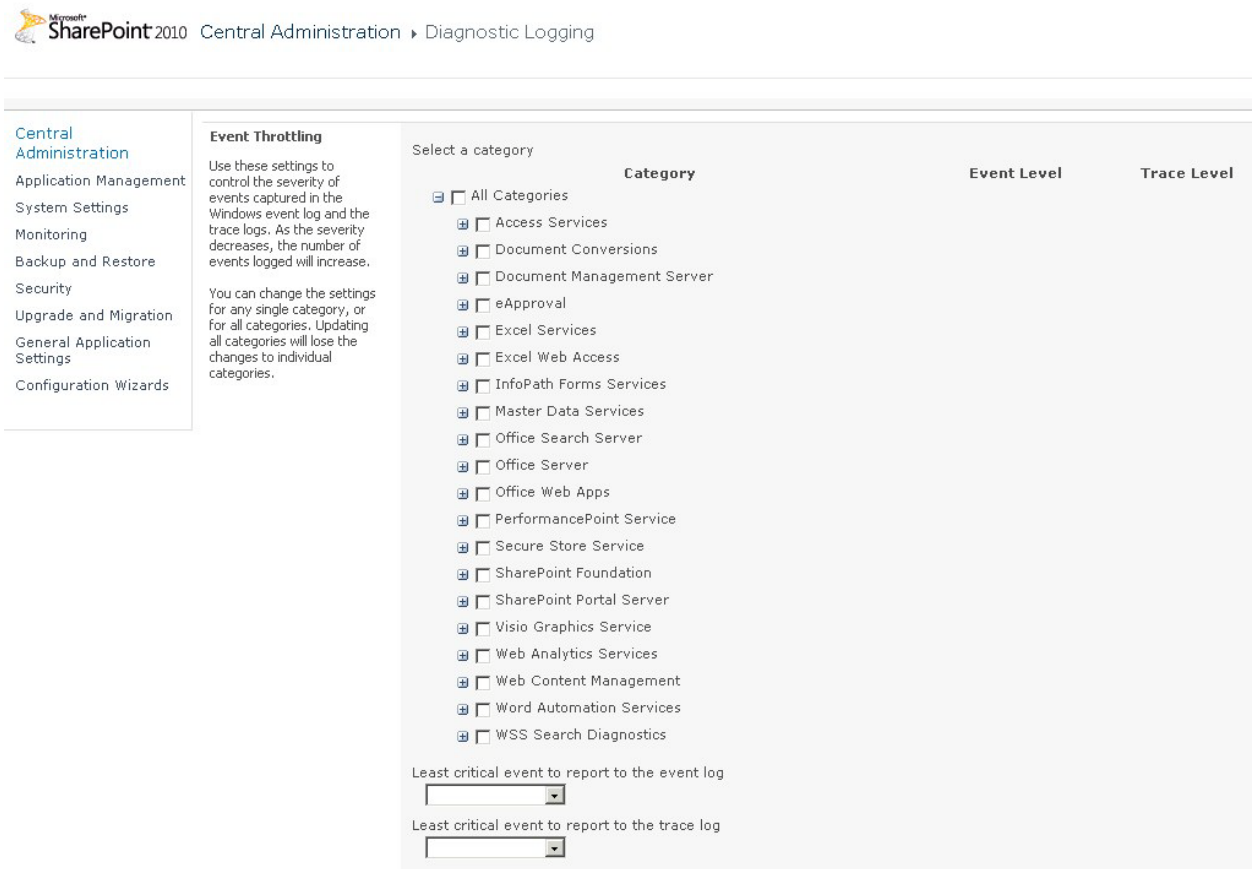

Figure 12: Diagnostic Logging

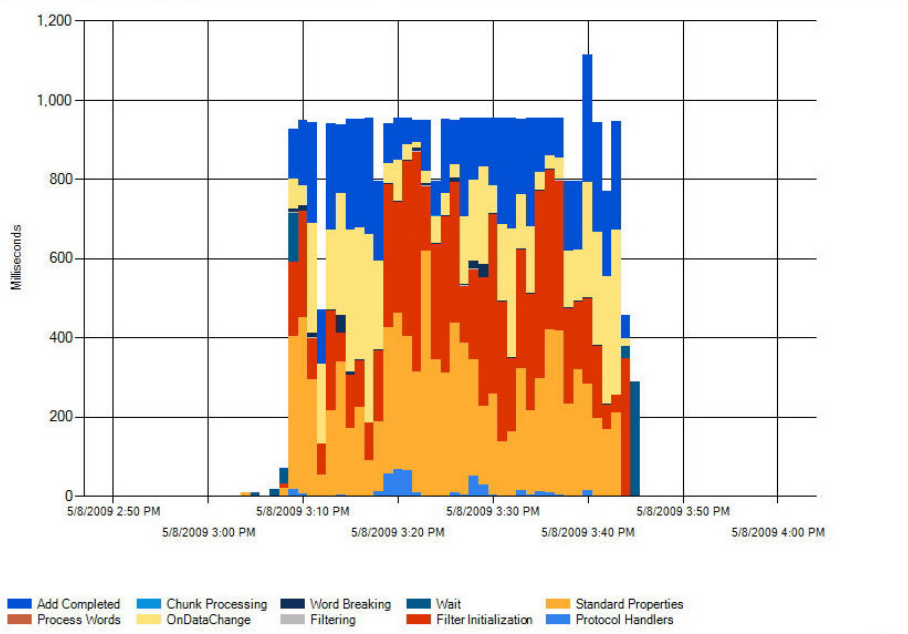

Figure 13: Crawl Activity Report

Diagnostics

The Unified Logging Service (ULS) includes improvements to manageability, log file improvements, correlation ID tracing, and Windows PowerShell™ scripting. Event throttling is one of the enhancements to manageability. It is configured on the Diagnostics Logging page illustrated earlier. Event throttling controls the severity of events that are captured in the Windows® event log and the trace logs. As the severity decreases, the number of events logged will increase. Events are categorized, and the administrator can change the settings for any single category or for all categories. Updating all categories will lose the changes to individual categories. Event Log Flood Protection (EVFP) can also be enabled on the same Web page. When EVFP is enabled, repeating events are detected and suppressed until conditions return to normal. The ULS now contains all application log events, and third-party logging software can be integrated into the ULS. The logs have been reduced in size by more than 50 percent by using the NTFS file system compression. The trace log can also be located in a specified location, and the storage duration and amount of storage can also be configured. Correlation IDs that are associated with each request help troubleshoot errors related to the request. SQL Profiler traces will also show correlation IDs to further assist in the

troubleshooting. SharePoint Server 2010 includes several Windows PowerShell cmdlets for retrieving information and configuring the ULS.

SharePoint Server 2010 logs feature usage and performance information into the usage database. This logging is done by the usage service application, and is enabled by default. Administrators can read, query, and build reports directly from the usage database because the schema is public. As such, third-party applications can also write their data to the usage database. The types of events that are logged are shown in the figure below and include: page requests, feature use, search query usage, site inventory usage, timer jobs, and rating usage. Because the usage database tracks feature usage and licensing information, the administrator can track the use of enterprise features against the number of enterprise client access licenses purchased, to help ensure compliance.

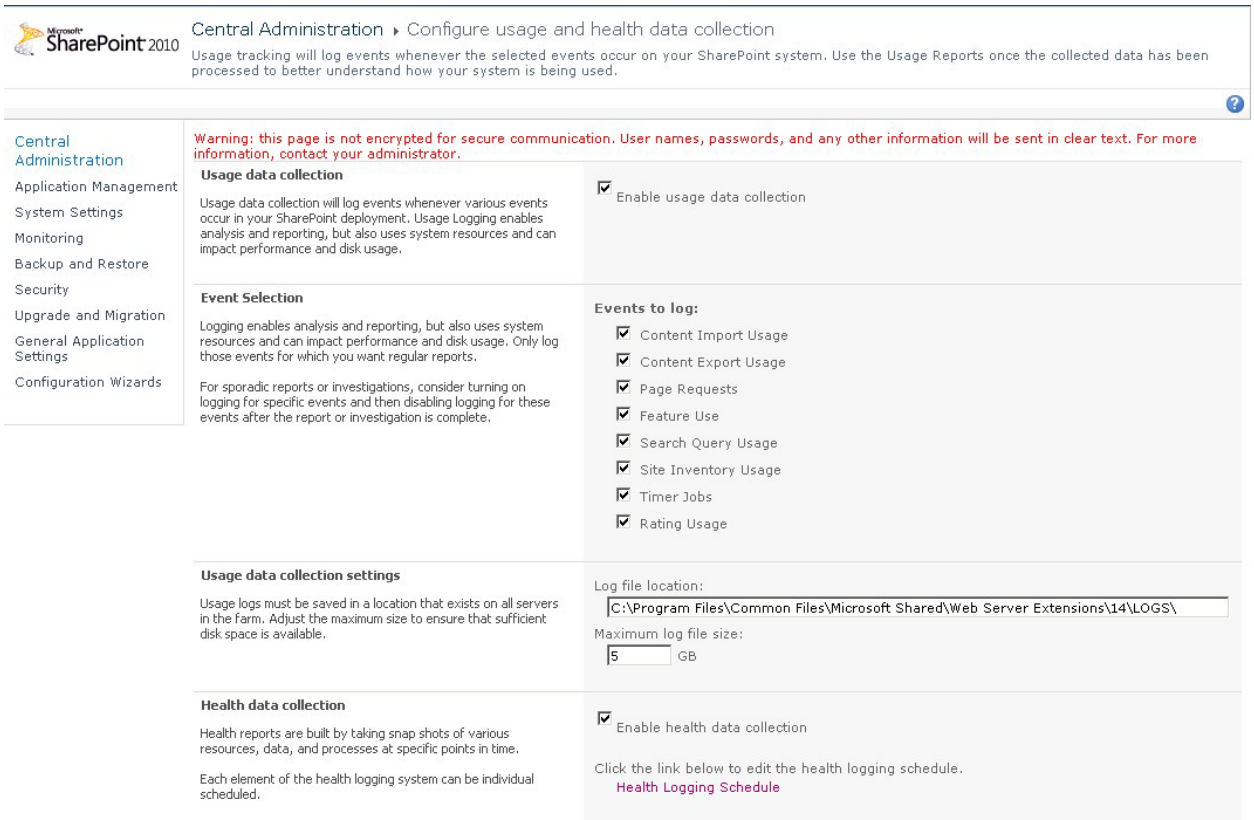
SharePoint 2010
Central Administration ▸ Configure usage and health data collection
Usage tracking will log events whenever the selected events occur on your SharePoint system. Use the Usage Reports once the collected data has been processed to better understand how your system is being used.
Central Administration
Application Management
System Settings
Monitoring
Backup and Restore
Security
Upgrade and Migration
General Application Settings
Configuration Wizards
Warning: this page is not encrypted for secure communication. User names, passwords, and any other information will be sent in clear text. For more information, contact your administrator.
Usage data collection
Usage data collection will log events whenever various events occur in your SharePoint deployment. Usage Logging enables analysis and reporting, but also uses system resources and can impact performance and disk usage.
Enable usage data collection
Event Selection
Logging enables analysis and reporting, but also uses system resources and can impact performance and disk usage. Only log those events for which you want regular reports.
For sporadic reports or investigations, consider turning on logging for specific events and then disabling logging for these events after the report or investigation is complete.
Events to log:
Content Import Usage
Content Export Usage
Page Requests
Feature Use
Search Query Usage
Site Inventory Usage
Timer Jobs
Rating Usage
Usage data collection settings
Usage logs must be saved in a location that exists on all servers in the farm. Adjust the maximum size to ensure that sufficient disk space is available.
Log file location:
C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\LOGS\
Maximum log file size:
5
GB
Health data collection
Health reports are built by taking snap shots of various resources, data, and processes at specific points in time.
Each element of the health logging system can be individual scheduled.
Enable health data collection
Click the link below to edit the health logging schedule.
Health Logging Schedule

Figure 14: Usage Logging

A new addition to server diagnostics is the developer dashboard. This dashboard displays detailed information for each page load and therefore helps troubleshoot

performance issues. This dashboard is disabled by default and can be enabled for each Web application independently through the use of Windows PowerShell.

Reliability and Monitoring

A new addition includes the SharePoint Maintenance Engine (SPME). The SPME is a new and improved version of the Best Practices Analyzer. It periodically, or on an on-demand basis, checks the administrative configuration, performance, best practices, and security issues, and makes recommendations to resolve potential issues.

Available only to users who have farm administrator credentials, the results are accessible through Central Administration via the Review Problems and Solutions link in the Monitoring section and the Health Status section, and they are also available in SCOM. The SPME maintains a list of rules called Health Rule Definitions. This list is created by the system to help ensure that the SharePoint environment is properly configured and healthy. An example of a rule definition is shown below.

Figure 15: Health Rule

SharePoint Server 2010 will ship a management pack for SCOM. It provides real-time alerts and troubleshooting in the context of larger infrastructure. It watches events, monitors performance counters, and takes corrective action where necessary. The management pack for SharePoint Server 2010 understands and discovers topology;

grabs events from ULS, the Windows Event Log, usage database, and SPME; and is integrated with ULS.

Remote BLOB Storage

For farms with large storage needs Microsoft® SharePoint® Server 2010 has implemented remote BLOB storage (RBS). With RBS, it is possible to have BLOBs from the all_docs table and customized pages in the database stored in a location outside of the database, such as on file system, allowing for a reduction in the amount of Microsoft SQL Server™ database storage. To use RBS, a provider is necessary. The RBS provider architecture is completely pluggable, meaning that third-party vendors can create and sell their own RBS providers and deliver specific capabilities. For example, a third party backup vendor could use an RBS provider as part of their backup and restore capability.

The RBS architecture provides another degree of freedom for the storage of content. Its intention is to be used natively by SharePoint during the upgrade process and by third party vendors that provide storage benefits as a service.

Performance Controls

Microsoft® SharePoint® Server 2010 provides two different performance controls that can be configured to help increase server performance and protect server resources during peak usage times: throttling and list controls. Throttling provides a way to control server resources and is designed so that the server can be protected from overload during peak business hours. Large-list settings provide a way to limit the different queries within a list to ensure that performance is not degraded when users query a list that contains many items.

Throttling

Through throttling settings, SharePoint Server 2010 provides a way for administrators to determine the level at which the server will go into throttling mode. Every five seconds, a job runs that checks server resources compared to configured levels. By default, Server CPU, Memory, Request in Queue, and Request Wait Time are monitored. After three unsuccessful checks, the server enters a throttling period and will remain in this state until a successful check is completed. Requests that were generated prior to the server's entering throttling mode will be completed. This will, in theory, keep users from losing

any current work when the server begins to throttle requests. Any new HTTP GET and Search Robot requests will generate a 503 error message and will be logged in the event viewer. While the server is in a throttling period, no new timer jobs will be started.

Throttling can be configured per Web application and is enabled through the Central Administration user interface or through a Windows PowerShell™ command. After throttling is enabled, the Windows PowerShell commands **Get-SPWebApplicationHttpThrottlingMonitor** and **Set-SPWebApplicationHttpThrottlingMonitor** are used to view and adjust the throttling thresholds. The image below is an example of the **Get-SPWebApplicationHttpThrottlingMonitor** Windows PowerShell command.

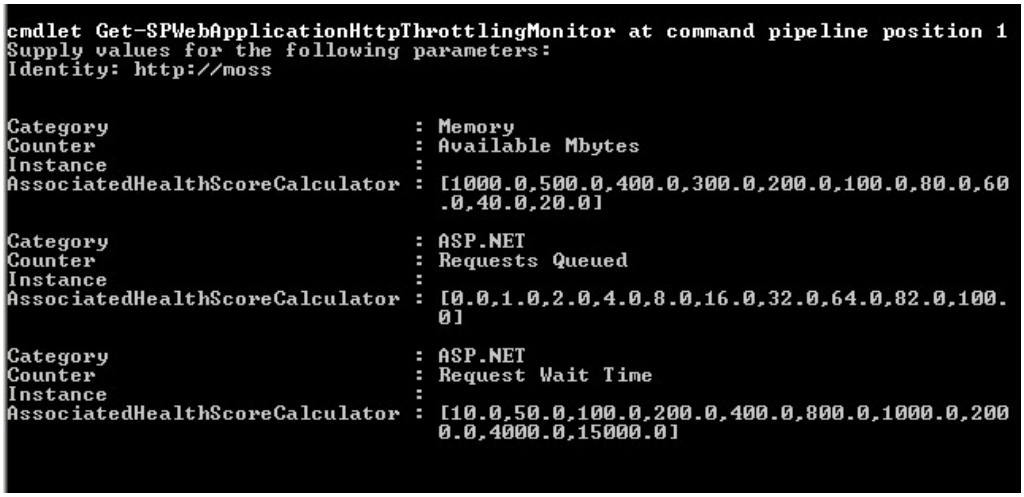

Figure 16: Windows PowerShell Commands for Monitoring

Controlling Large List Activities

SharePoint Server 2010 has the ability to support up to 50 million items within a SharePoint list; however, as lists grow in size they can become increasingly slow to return queries that include many results. To provide users with consistent server response, SharePoint Server 2010 provides several different settings that will restrict the queries that can be run against a large list. These settings can be configured for each Web application and will control the following:

- The number of results that can be returned in a user query. The default value is 5,000 items.

- The warning level for site administrators that their list is close to the threshold limit. This warning is displayed on the list settings page. The default value is 3,000 items.
- The number of query results that can be displayed for a site administrator. This allows site administrators to access a larger data set than other site users. The default value is 20,000 items.
- The maximum number of unique scopes in a list. The default value is 50,000 items.
- The number of lookup columns that each list can include. The default value is 6 lookup columns.

These settings are configured for each Web application. They can be accessed through the General Settings option on the Web application ribbon.

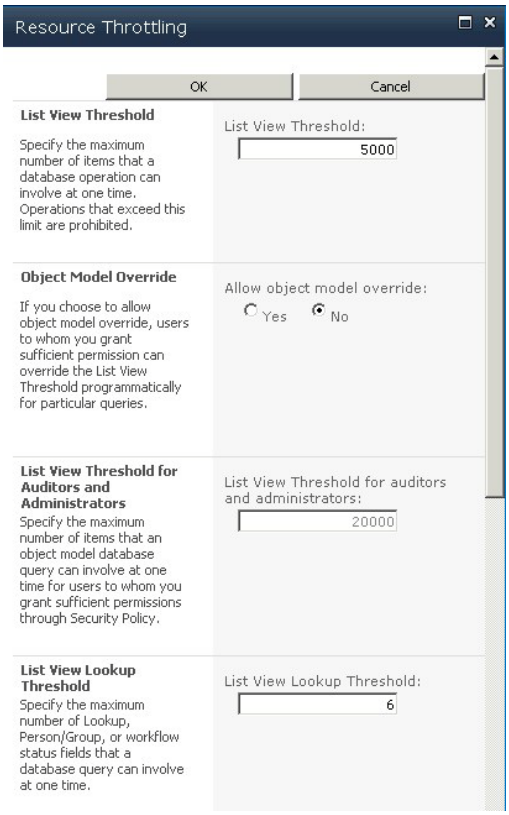
Resource Throttling
OK
Cancel
List View Threshold
Specify the maximum number of items that a database operation can involve at one time. Operations that exceed this limit are prohibited.
List View Threshold:
5000
Object Model Override
If you choose to allow object model override, users to whom you grant sufficient permission can override the List View Threshold programmatically for particular queries.
Allow object model override:
Yes
No
List View Threshold for Auditors and Administrators
Specify the maximum number of items that an object model database query can involve at one time for users to whom you grant sufficient permissions through Security Policy.
List View Threshold for auditors and administrators:
20000
List View Lookup Threshold
Specify the maximum number of Lookup, Person/Group, or workflow status fields that a database query can involve at one time.
List View Lookup Threshold:
6

Figure 17: Resource Throttling

Although the settings described above apply to all lists within the Web application, additional settings are available and can be configured to override the Web application settings:

- Allowing object model override will allow for a query to be programmatically written against a list where the query threshold is not applied.
- Enabling a daily time window for large queries will provide a way for administrators to configure a specific window of time where the query thresholds are not applied.
- A Windows PowerShell command that will restrict specific lists from being queried through the object model (this would cancel out the object model override mentioned above for a specific list instance).

Windows PowerShell Administration

Windows PowerShell™ is the new command-line interface and scripting language specifically designed for system administrators that will be used for Microsoft® SharePoint® Server 2010 administration. Although both Cmd.exe and Stsadm.exe environments will be maintained for backward compatibility, SharePoint Server 2010 command-line administration will use Windows PowerShell. For practical purposes, you can view Windows PowerShell as complementing Cmd.exe in the Windows administration context and superseding Stsadm.exe for SharePoint administration. You might ask yourself, What's so unique about Windows PowerShell? Well, unlike most command-line systems that accept and return text, Windows PowerShell supports and uses XML and objects. XML and object support is a pretty major change that brings entirely new tools and methods to greatly improve control, efficiency, and productivity for developers and administrators. Windows PowerShell is built on top of the .NET common language runtime (CLR) and the .NET Framework. Windows PowerShell is not an updated version of Cmd.exe or MS-DOS®; it's a complete command-line system that is far more powerful than any previous Windows-based command-line prompt environment.

Cmdlets

Windows PowerShell introduces the concept of a cmdlet (pronounced "command-let"). A cmdlet is a single function that can be used separately or combined with other cmdlets to perform complex tasks and automate administration. Windows PowerShell provides a set of pre-built cmdlets, fully supports cmdlet customization to suit your environment, and supports various customizable Windows PowerShell providers that make available specific sets of cmdlets.

A cmdlet is a simple command used for interaction with any managed application, including the operating system. It is analogous to such shell commands as Cmd.exe, but the processing of Windows PowerShell commands is very different. The traditional shell generally processes commands as separate executable programs. Each program has to parse the input, distinguish between positional and named parameters, bind values to the correct parameters, format the output, and display the output. In contrast, Windows PowerShell processes commands as instances of a .NET class, focusing on the simple cmdlet model. You provide the parameters and validate the values, and then furnish details of object types and formatting. Windows PowerShell does the rest of the work: parsing the parameters, binding them to their values, formatting the output, and displaying the output.

Cmdlet names are comprised of *verbs* and *nouns*, which denote their functional properties. For example, the cmdlet name **Get-SPSite** combines the verb (command) **Get** with the noun (object) **SPSite** to name the cmdlet that retrieves a specified SharePoint **SPSite** object or collection. Cmdlet nouns take parameters as name-value pairs that give specificity to the cmdlet noun; when invoked, the cmdlets return output objects. The returned objects, in turn, have properties that are displayed as name-value pairs.

The figure below shows the Windows PowerShell command-line window when the **Get-SPSite** cmdlet has been invoked. The cmdlet uses a URL parameter in the form `–URL http://localhost` to identify the specific **SPSite** object it will retrieve. The cmdlet then provides an equivalent name-value item that identifies the return object.

A cmdlet is not an executable; it is an instance of a .NET Framework class and therefore it returns objects rather than a text stream. These objects can be passed (or "piped") to another cmdlet as input, in sequence. In this way, cmdlets can be chained together, providing enormous flexibility.

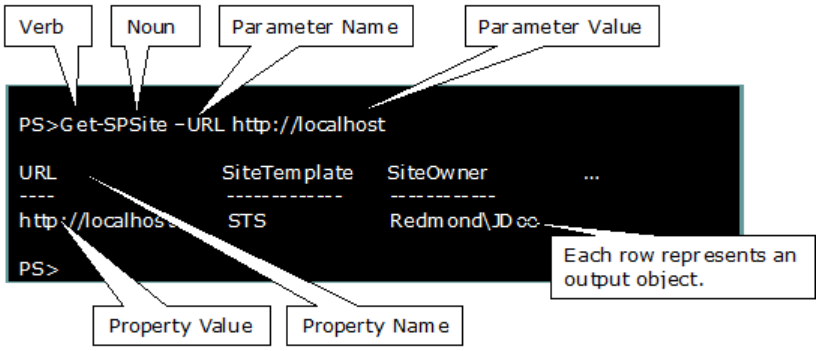

Figure 18: Windows PowerShell command

As you can see, Windows PowerShell is not just a new command shell. It is also, and perhaps more importantly, a new scripting language. Windows PowerShell is installed natively with over 100 core Windows cmdlets. The library of SharePoint Server 2010 cmdlets, which presently number over 300, will be installed on top of these core cmdlets. The following two sample scenarios describe how an administrator might use Windows PowerShell in SharePoint Server 2010 to help manage work efficiently.

To get a list of all cmdlets, you can use the built-in Windows PowerShell cmdlet **Get-Command**. First, open the SharePoint 4.0 Management Console from the Start menu under Administrative Tools. Type the following at the prompt and press ENTER:

```
Get-Command –noun sp*
```

After you have the complete list, you can page through it and look for commands that interest you. After you find a command you would like more information about, you can use the built-in Help for additional information. Type the following at the prompt and press ENTER:

```
Get-Help <cmdlet>
```

This will give you the simple help. You can also add a **–full** or **–detailed** string to the end of the command to get additional information.

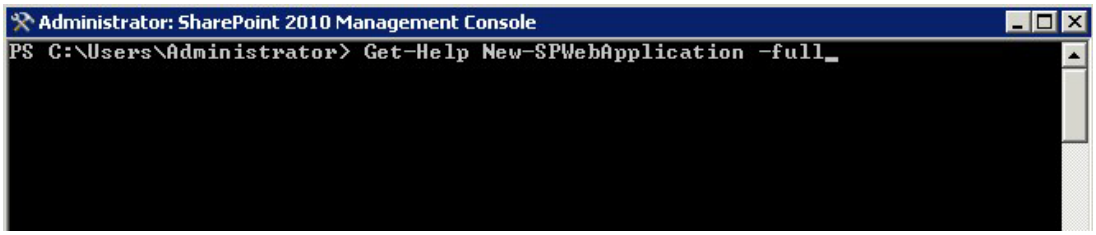

Figure 19: Show Help Content for Windows PowerShell Cmdlets

Scenario: Provisioning of Managed Accounts

Steve has just completed the installation of SharePoint Server 2010 and is preparing to configure the farm. He has put in a request to the Active Directory administrators to have all necessary service accounts provisioned. All of these accounts were created by using the same password. Steve now needs to set up all of these accounts as managed accounts (this will be covered in more detail later in this guide). This way, SharePoint Server can change their initial passwords to something that only SharePoint Server knows, and SharePoint Server can automatically change the password in the future as required by domain policy. This will greatly reduce the password management load on Steve, because after he configures the accounts as managed he is no longer in control of their passwords. SharePoint Server 2010 will handle everything.

Steve begins by creating a text file that includes one account per line for all of the accounts he needs to configure. He names the file Usernames.txt. Steve can then run the following Windows PowerShell command.

```
$password = Read-Host "What is the password for all accounts?" -assecurestring ; Get-Content usernames.txt | ForEach-Object {New-SPManagedAccount -password $password -username litwareinc\$_; Set-SPManagedAccount -identity $_ -autogenerate -confirm:$False}
```

To better understand what Steve is doing, look at the command in the pieces that make up the whole.

```
$password = Read-Host "What is the password for all accounts?" -assecurestring ;
```

The **Read-Host** cmdlet causes the user to be prompted to enter a password. For Steve, this is the password the Active Directory administrator originally configured for all of the accounts. The password is then saved to the variable *$password* as a secure string. The

dollar sign ($) denotes a variable in Windows PowerShell. The semicolon (;) means "You are done running this cmdlet, process the next cmdlet." This is the same as starting a new line in Windows PowerShell.

```
Get-Content usernames.txt
```

This cmdlet reads each line of the file Usernames.txt and stores them as objects.

```
|
```

The pipe symbol means "Send the output of the preceding cmdlet to the next cmdlet."

```
ForEach-Object {New-SPManagedAccount -password $password -username litwareinc\$_; Set-SPManagedAccount -identity $_ -autogenerate -confirm:$False}
```

The **ForEach-Object** cmdlet tells Windows PowerShell to run all cmdlets inside the braces ({ }) on each object separately until it processes all objects. In Steve's case, this means "Take each username, one at a time, and process them."

```
New-SPManagedAccount -password $password -username litwareinc\$_;
```

The first step in that process is to define each account as a managed account. The **New-SPManagedAccount** is passed the *$password* variable Steve entered earlier, and for the user name it uses the current username (object) from the text file. This is represented by the variable $_ which Windows PowerShell automatically created when you got the objects from the text file. Finally the semicolon (;) is used to say "This cmdlet is done, move to the next."

```
Set-SPManagedAccount -identity $_ -autogenerate -confirm:$False
```

This cmdlet takes the created managed account of $_ and automatically generates a new password for it. The **–confirm:$False** string suppresses the command from prompting Steve to confirm the password change.

Now Steve has configured all of his managed accounts and can use them for such things as configuring service applications and application pool identities without regard to their actual passwords.

Windows PowerShell Out-Performing Stsadm Scenario

Nicole needs to activate a feature across 5,500 site collections. By using Windows PowerShell to run the Stsadm command, she came up with the following script to automate the work:

```
foreach ($site in (get-spsite -limit 5000000 -Webapplication $WebApp))
{               Write-Host "Activating feature " $solutionName "on" $site.url "...";
            Write-Host stsadm "-o activatefeature -url" $site.url "-filename" $featureFileName;
            stsadm -o activatefeature -url $site.url -filename $featureFileName
            if( $lastexitcode -ne 0 )
            {               Write-Host "Something went wrong activating the site feature. Exit code: " $lastexitcode "`n"  -ForegroundColor Red;
                        $failure = $true;
                        $error.Clear();
            }
}
```

When she ran the script in her test environment, she found it took more than 12 hours to run. Nicole then decided to look for a better way of doing things. After talking to Grant, she found out there was built-in cmdlet to do this exact task. She then tried this script:

```
Get-SPSite –Limit ALL –WebApplication $WebAppNameorUrl |%{ Enable-SPFeature $FeatureIdOrName –url $_.Url }
```

The script ran in less than one hour. Lesson learned: If Stsadm.exe can do the operation, Windows PowerShell can do it too, and generally more efficiently.

Managed Accounts

To reduce the load of managing various service accounts in Microsoft® SharePoint® Server 2010, the concept of managed accounts has been introduced. Much like managed accounts in Windows Server® 2008, they allow SharePoint Server to take control of all the service accounts you use. After SharePoint Server has control of these accounts, it can either manage their passwords — automatically changing them as necessary — or it can notify you when an accounts password is about to expire, allowing

you to make the change yourself. This functionality will greatly reduce the burden on the SharePoint farm administrator who is responsible for maintaining these various accounts and should prevent unnecessary downtime due to expired accounts.

Backup and Restore

Microsoft® SharePoint® Server 2010 provides a broad range of levels for performing backups, including the entire farm, farm configuration information, site collections, subsites, or lists. Backups can be done by using the Central Administration pages or Windows PowerShell™. Using the backup and restore features available in SharePoint Server 2010 gives SharePoint administrators a powerful set of tools that allow them to manage their environment with minimal downtime and operational costs.

> **Note** Stsadm is still available to perform backups to maintain backward compatibility. It is recommended that all new backup plans incorporate Windows PowerShell in place of Stsadm.

Central Administration vs. Windows PowerShell

SharePoint Server 2010 uses two different tools to configure backup and recovery. Central Administration provides a user interface where SharePoint Administrators will be prompted via menu structures to select the information that needs to be backed up.

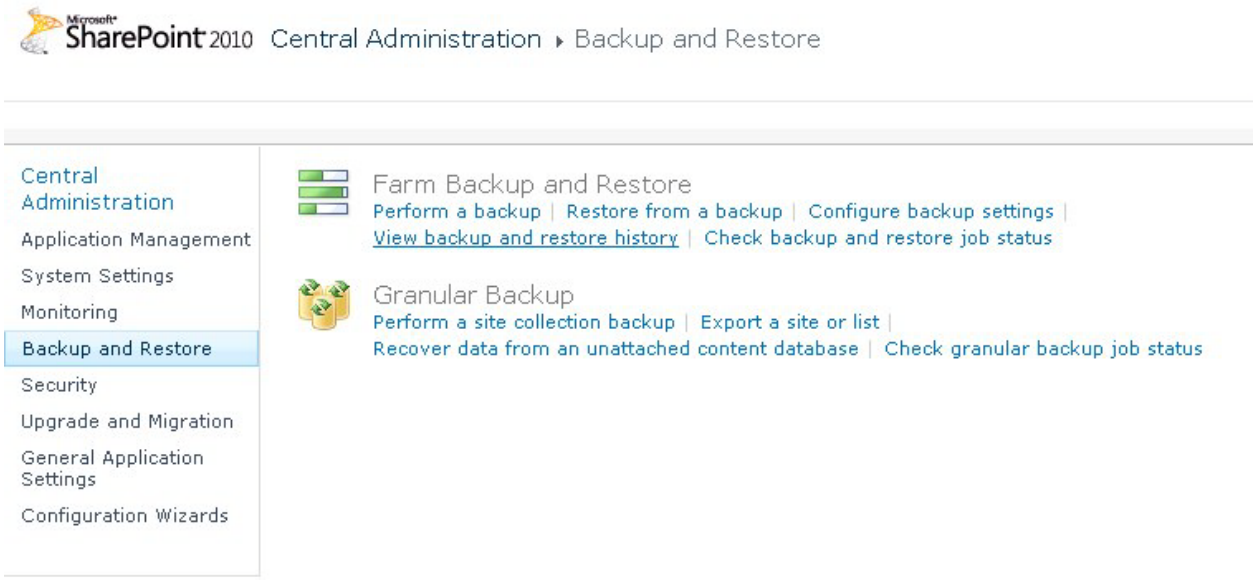
SharePoint 2010
Central Administration ▸ Backup and Restore
Central Administration
Application Management
System Settings
Monitoring
Backup and Restore
Security
Upgrade and Migration
General Application Settings
Configuration Wizards
Farm Backup and Restore
Perform a backup | Restore from a backup | Configure backup settings |
View backup and restore history | Check backup and restore job status
Granular Backup
Perform a site collection backup | Export a site or list |
Recover data from an unattached content database | Check granular backup job status

Figure 20: Backup and restore user interface

Windows PowerShell is a command line tool that provides SharePoint administrators a way to perform backup and recovery with additional options such as file compression or working with SQL snapshots. Listed below are a few of the benefits available when working with Windows PowerShell:

- Windows PowerShell scripts can be developed and scheduled (with Windows Task Scheduler), whereas Central Administration is used for single-use backups and restores.
- Windows PowerShell has the advantage of running against SQL snapshots instead of the production database. One of the parameters of the Windows PowerShell command will cause a SQL snapshot to be generated, and then Windows PowerShell will run the action against the snapshot instead of the production database. This will reduce the resource impact of the backup operation on the production environment.
- With Windows PowerShell, SharePoint administrators will have more granular control of options for the backup or restore.

The remainder of this section will identify when additional features are made available through Windows PowerShell commands that are not available through the standard user interface.

For more information about Windows PowerShell, see the Windows PowerShell section of this guide and refer to the following link: Server Management (http://go.microsoft.com/fwlink/?LinkId=156748).

Granular Backup and Restore

SharePoint Server 2010 provides several new features that provide a granular level of backup for various components of site content. This includes content at the site, subsite, and list level. This level of granular backup provides SharePoint administrators with the ability to quickly and flexibly backup and restore content to meet user demand.

Back Up a Site Collection

Backing up a site collection will include the top level site and all subsites. When backing up the site collection through Central Administration, a SharePoint administrator identifies the site collection and provides a destination for the storage of the backup file.

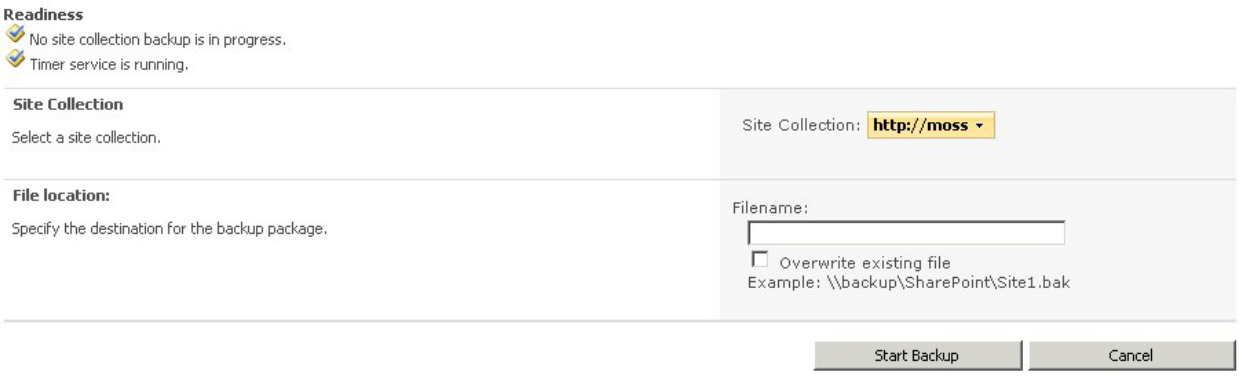

Figure 21: Back up a site collection

The same operation can be completed by using the **Backup-SPSite** Windows PowerShell command. Note that additional parameters are available, including using a SQL snapshot.

```
SYNTAX
    Backup-SPSite -Identity <SPSitePipeBind> -Path <String> [-AssignmentCollect
    ion <SPAssignmentCollection>] [-Confirm [<SwitchParameter>]] [-Force <Switc
    hParameter>] [-NoSiteLock <SwitchParameter>] [-UseSqlSnapshot <SwitchParame
    ter>] [-WhatIf [<SwitchParameter>]] [<CommonParameters>]
```

Figure 22: Windows PowerShell command for backing up a site

To restore the site collection, the **Restore-SPSite** Windows PowerShell command is used.

Export a Site or List

Through Central Administration a SharePoint Administrator can configure a backup of a subsite or list. As seen in the image below, after the site is selected, there are additional options for the site and for specific lists. From this page, the administrator can also choose to export security and select the different versions that will be exported with the site or list.

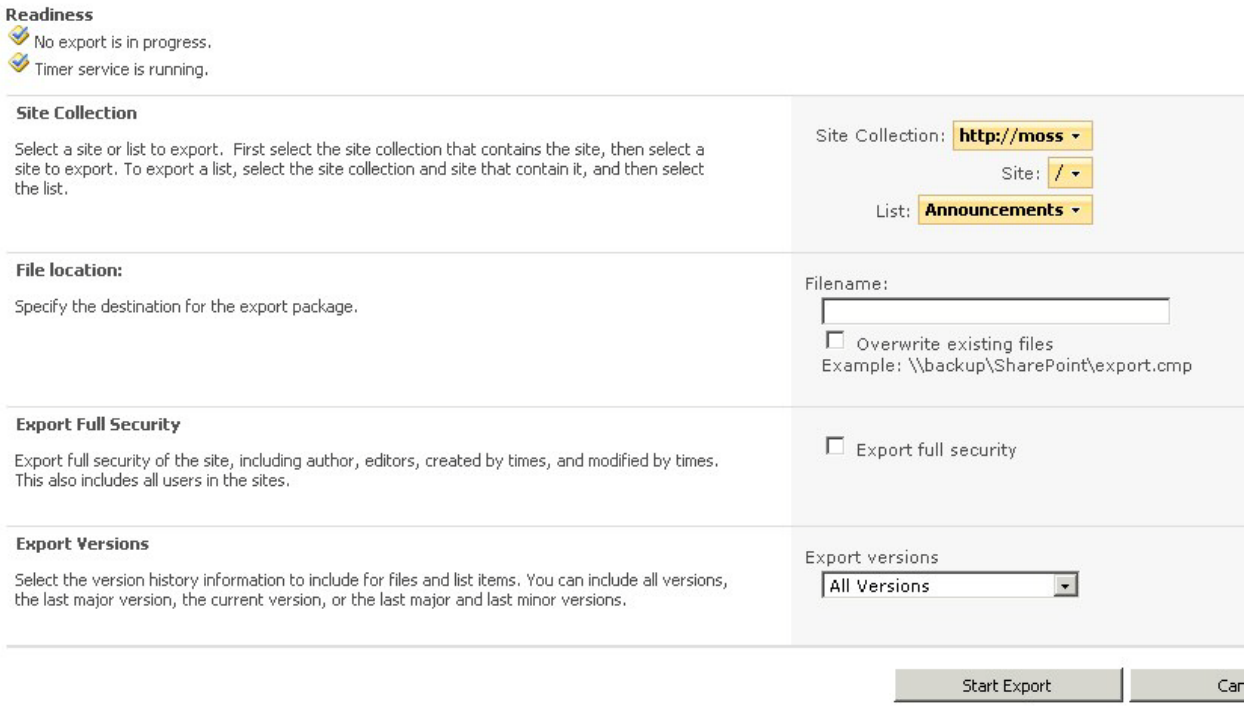
Readiness
No export is in progress.
Timer service is running.
Site Collection
Select a site or list to export. First select the site collection that contains the site, then select a site to export. To export a list, select the site collection and site that contain it, and then select the list.
Site Collection: http://moss
Site: /
List: Announcements
File location:
Specify the destination for the export package.
Filename:
Overwrite existing files
Example: \\backup\SharePoint\export.cmp
Export Full Security
Export full security of the site, including author, editors, created by times, and modified by times. This also includes all users in the sites.
Export full security
Export Versions
Select the version history information to include for files and list items. You can include all versions, the last major version, the current version, or the last major and last minor versions.
Export versions
All Versions
Start Export
Car

Figure 23: Export a site or list in the user interface

Similar to the site collection backup, Windows PowerShell provides some additional parameters, including the use of SQL snapshots, compression, and logging.

```
SYNTAX
    Export-SPWeb -Identity <SPWebPipeBind> -Path <String> [-AssignmentCollectio
    n <SPAssignmentCollection>] [-CompressionSize <Int32>] [-Confirm [<SwitchPa
    rameter>]] [-Force [<SwitchParameter>]] [-HaltOnError [<SwitchParameter>]] [-Ha
    ltOnWarning [<SwitchParameter>]] [-IncludeUserSecurity [<SwitchParameter>]] [-I
    ncludeVersions <LastMajor | CurrentVersion | LastMajorAndMinor | All>] [-It
    emUrl <String>] [-NoFileCompression [<SwitchParameter>]] [-NoLogFile [<SwitchP
    arameter>]] [-UseSqlSnapshot [<SwitchParameter>]] [-WhatIf [<SwitchParameter>]
    ] [<CommonParameters>]
```

Figure 24: Export a site or list in Windows PowerShell

To restore the site or list, the **Import-SPWeb** Windows PowerShell command is used.

Recover Data from an Unattached Content Database

In SharePoint Server 2010, SharePoint administrators will be able to restore site collections, sites, and lists from unattached content databases. What does this mean? SharePoint administrators can simply do a SQL Server restore of the content database to any computer running SQL Server, then tell SharePoint to connect to it. Now they can browse the contents of the database and then back up or export the content they need. This eliminates the need to build a second farm for granular recovery. After the backup

or export is finished, they can the restore that content to the appropriate place in their SharePoint site and delete the restored database.

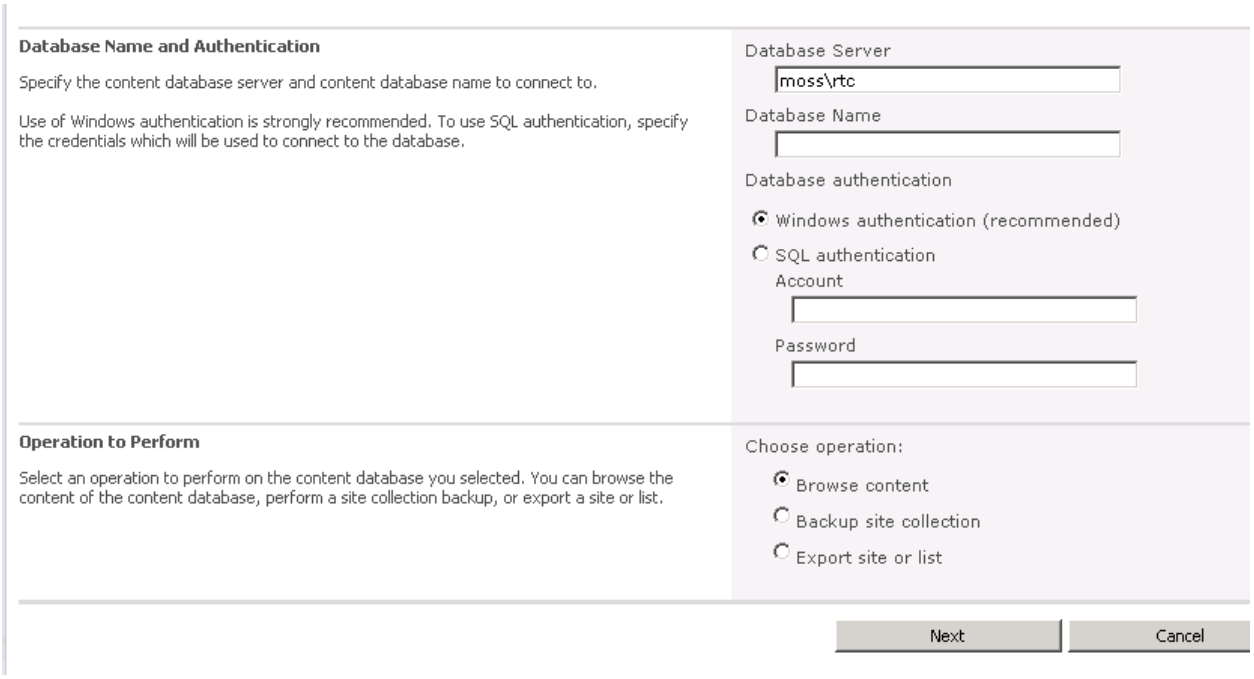
Database Name and Authentication
Specify the content database server and content database name to connect to.
Use of Windows authentication is strongly recommended. To use SQL authentication, specify the credentials which will be used to connect to the database.
Database Server
moss\rtc
Database Name
Database authentication
Windows authentication (recommended)
SQL authentication
Account
Password
Operation to Perform
Select an operation to perform on the content database you selected. You can browse the content of the content database, perform a site collection backup, or export a site or list.
Choose operation:
Browse content
Backup site collection
Export site or list
Next
Cancel

Figure 25: Recover data from an unattached database

After you point to the unattached content database you will be able to browse, back up, or export the content. The same level of backup granularity is available for both attached and unattached content databases. The process for selecting the content and options for backup will be the same as above, which is illustrated below.

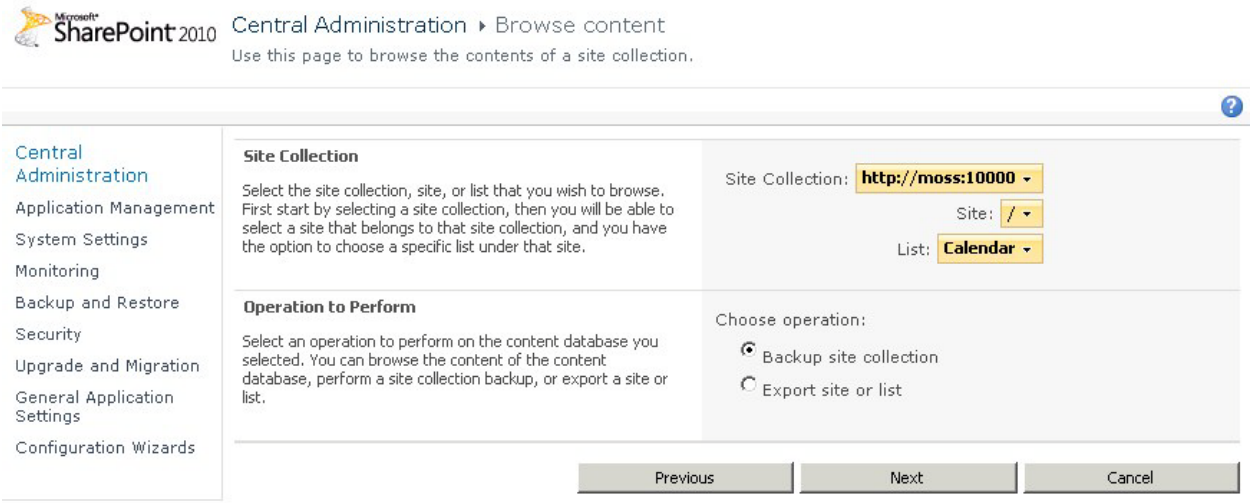
SharePoint 2010
Central Administration ▸ Browse content
Use this page to browse the contents of a site collection.
Central Administration
Application Management
System Settings
Monitoring
Backup and Restore
Security
Upgrade and Migration
General Application Settings
Configuration Wizards
Site Collection
Select the site collection, site, or list that you wish to browse. First start by selecting a site collection, then you will be able to select a site that belongs to that site collection, and you have the option to choose a specific list under that site.
Site Collection: http://moss:10000
Site: /
List: Calendar
Operation to Perform
Select an operation to perform on the content database you selected. You can browse the content of the content database, perform a site collection backup, or export a site or list.
Choose operation:
Backup site collection
Export site or list
Previous
Next
Cancel

Figure 26: Browse content from an unattached database

Scenario: Using Windows PowerShell to Script Backing Up All Site Collections Individually

Jennifer is an overly cautious SharePoint administrator who doesn't have complete faith in the database backups that are being performed each night by her database administrators. She would prefer that she had her own backups, just in case. Looking at her options, she can do a full farm backup by using the user interface, but she doesn't want to VPN in every night at midnight to start the job — so that is out. She could set up a Windows PowerShell script to do the farm backup each night by using **Backup-SPFarm** and then schedule that by using Task Scheduler — but that backup is best served for disaster recovery and that is what the database administrators are protecting. So she settles on using a Windows PowerShell script that will take advantage of the cmdlet **Backup-SPSite** to back up individual site collections. This will give her a great deal of flexibility for recovery. The script she used is:

```
Get-SPWebApplications | Get-SPSite | ForEach-Object {$FilePath = "c:\backups\" + $_.Url.Replace(http://,"").Replace("/","-" + ".bak"); Backup-SpSite –Identity $_ -path $Filepath
```

Farm Backup and Restore

SharePoint Server 2010 has built in capabilities for backing up the entire farm to provide recovery for a catastrophic failure. This interface can be accessed from within Central Administration for ad hoc backups and recovery, or it can be scripted by using the Windows PowerShell cmdlets **Backup-SPFarm** and **Restore-SPFarm**, respectively. A new addition to the catastrophic backup scenario is the capability to do a configuration-only backup.

The configuration-only backup outputs an XML file that can be used to restore all settings in the farm. This could be used for recovery of the farm or to build a different farm entirely by using the same settings. This is possible because all settings that are backed up are non–machine-specific.

Read-Only Databases

SharePoint Server 2010 has the ability to recognize read-only SQL databases. Whenever SharePoint Server is connected to a SQL database that has been placed in read-only mode, SharePoint Server will security trim all site collections in the database so that they are read-only to all users, removing all add and edit functionality from the sites, even for

administrators. This feature will provide the tools to help mitigate downtime during upgrades and server maintenance. For example, a secondary farm can be created and log shipping can be used to keep it current with the production environment. DNS can then be updated to point to the secondary environment, where all content databases are set to read-only. The production farm can be upgraded or updated without users accessing it. After it is complete and brought back online, DNS can be updated again to point to the production farm. Using this approach will allow users to have full read access to their data during the upgrade process.

SQL Mirroring

SharePoint Server 2010 supports SQL mirroring natively. This allows a farm administrator to configure the SQL environment to mirror a SharePoint content database. Whenever a new database is created, the SharePoint administrator will be prompted to enter the failover server. SharePoint will then communicate with the witness server and issues are discovered, SharePoint Server will fail over to the hot backup of the environment.

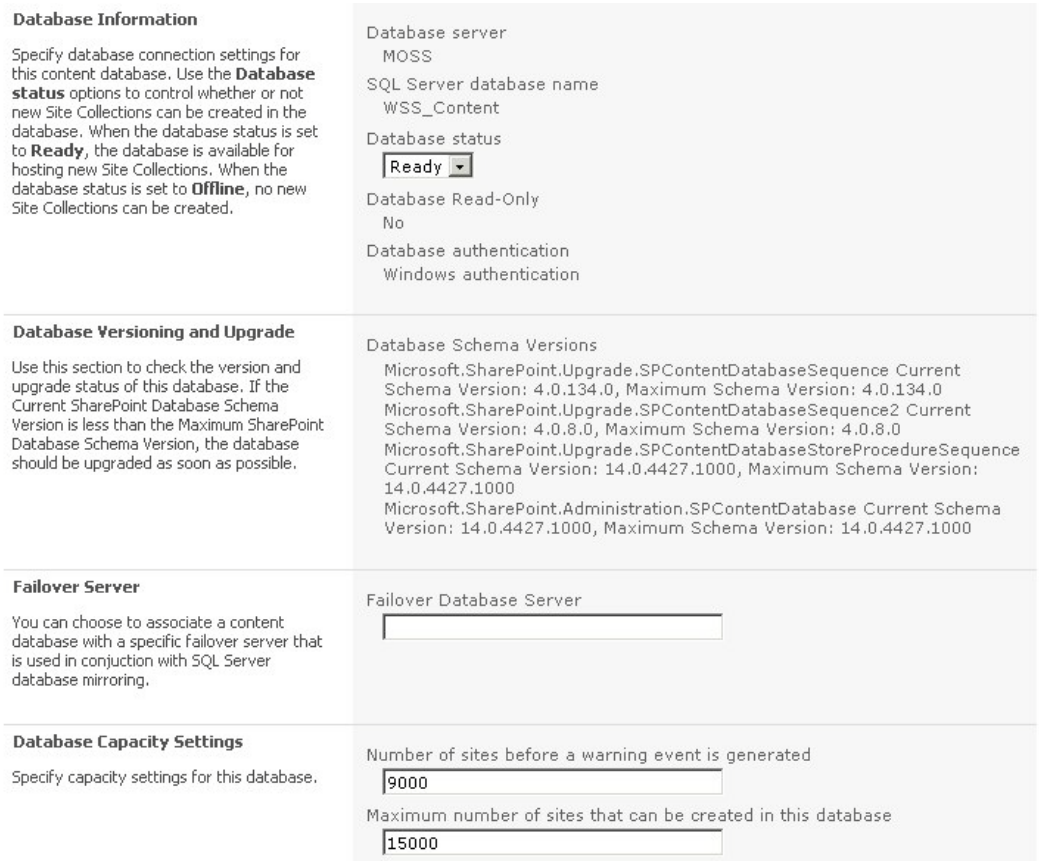
Database Information
Specify database connection settings for this content database. Use the Database status options to control whether or not new Site Collections can be created in the database. When the database status is set to Ready, the database is available for hosting new Site Collections. When the database status is set to Offline, no new Site Collections can be created.
Database server
MOSS
SQL Server database name
WSS_Content
Database status
Ready
Database Read-Only
No
Database authentication
Windows authentication
Database Versioning and Upgrade
Use this section to check the version and upgrade status of this database. If the Current SharePoint Database Schema Version is less than the Maximum SharePoint Database Schema Version, the database should be upgraded as soon as possible.
Database Schema Versions
Microsoft.SharePoint.Upgrade.SPContentDatabaseSequence Current Schema Version: 4.0.134.0, Maximum Schema Version: 4.0.134.0
Microsoft.SharePoint.Upgrade.SPContentDatabaseSequence2 Current Schema Version: 4.0.8.0, Maximum Schema Version: 4.0.8.0
Microsoft.SharePoint.Upgrade.SPContentDatabaseStoreProcedureSequence Current Schema Version: 14.0.4427.1000, Maximum Schema Version: 14.0.4427.1000
Microsoft.SharePoint.Administration.SPContentDatabase Current Schema Version: 14.0.4427.1000, Maximum Schema Version: 14.0.4427.1000
Failover Server
You can choose to associate a content database with a specific failover server that is used in conjuction with SQL Server database mirroring.
Failover Database Server
Database Capacity Settings
Specify capacity settings for this database.
Number of sites before a warning event is generated
9000
Maximum number of sites that can be created in this database
15000

Figure 27: Configure a content database for failover database server

SQL Snapshots

SharePoint Server 2010 backup and export take advantage of SQL snapshots. When a Windows PowerShell command is run by using a SQL snapshot the first thing that happens is that a snapshot of the database is created and then the remaining command is executed against the SQL snapshot. This provides a method that will reduce the load on the production databases. For more information about SQL snapshots, refer to the following link: Database Snapshots (http://go.microsoft.com/fwlink/?LinkId=156753).

Search and Index Backup and Restore

SharePoint Server 2010 has the ability to back up search and index databases by using a point and time approach. This will allow SharePoint Administrators to restore the search administration, crawl, and query databases. SharePoint Server will then be able to complete an incremental crawl based on all content that has been modified since the backup was taken. This will eliminate the requirement to complete a full crawl if the search database needs to be restored.

Multi-tenancy and Hosting

Microsoft® SharePoint® Server 2010 has the capability to isolate and separate data from different Web sites while sharing service application resources across these same sites. This capability is called *multi-tenancy*. Multi-tenancy of services creates a true hosting environment and makes it possible to share service resources across customers (tenants) while partitioning data based on *site subscriptions*. Site subscriptions group tenant data across all site collections owned by the tenant, and provide the ability to separate and group each tenant's data in an otherwise shared environment. Administrators can centrally deploy and manage features and services while giving tenants full control over the usage and experience.

Site Subscriptions

Multi-tenancy relies on site subscriptions and subscription IDs. Site collections for each tenant are grouped together by site subscription based on a common subscription ID. The subscription ID is used to map features and services to tenants and also to partition service data according to tenant. The following key points provide additional perspective about the capability:

- All site collections for a single site subscription must reside in the same Web application.
- Multiple site subscriptions can be hosted in a single Web application. Multiple site subscriptions can share the same database.
- Administrators define which services are available and activated for each tenant. The subscription ID for a tenant is used to map the service to the site collections.
- Multiple tenants can share service data. For example, multiple branches of an organization can share profile data. In this case, the multiple branches share a data partition within the profile service.
- Service data can be partitioned for each tenant, ensuring that customer data is not exposed to other tenants. In this case, data for a single tenant is contained within a separate data partition for the service.

Administration

Administrators can host multiple tenants on the same farm and centrally manage the deployment of services and features. Tenants can manage the configuration of administrator-delegated features and control the functionality of their site collections. Administrative functionality is aligned with common hosting roles, as summarized in the following table.

Role	Description
Hosting company	Manages the farm-level settings and hardware Controls database configurations Installs all new approved features and solutions Can brand the Tenant Administrator pages
Hosted company administrator	Purchases space, features, and bandwidth from hosting company Controls the architecture of customer sites but not the content Reviews usage statistics
Hosted company	Owns site collection

	Installs or removes features and solutions Configures features and services Reviews usage statistics

Tenants administer their sites by using a site template titled Tenant Administration, which is associated with a subscription ID. This site template is hidden and can be granted to tenants only by a farm administrator.

Branding

There have been a number of changes related to branding the user interface added to Microsoft® SharePoint® Server 2010.

Themes

In previous versions of SharePoint Server and SharePoint Foundation (known previously as Windows® SharePoint Services), themes were a common way to change the look and feel of the site. Administrators could go to the Site Settings page and select from one of several themes. Although this was a quick and easy way to make your site look different from the default site, often the themes available out of the box didn't match the corporate brand for a particular organization. In those cases, the options were either to create a custom theme, which required some development and modification of files on the server, or a company would just have to accept what was available.

SharePoint Server 2010 has changed the way themes work, making them easier to customize. Users who have created a custom theme in Microsoft Office PowerPoint® might notice that the interface to create themes looks very similar. In fact it is very similar! It is actually possible to import PowerPoint themes directly into SharePoint Server. Additionally, themes can now be applied to all subsites from this interface. The new themes functionality in SharePoint Server 2010 will make it faster and easier than ever before for administrators to create and deploy a customized user interface.

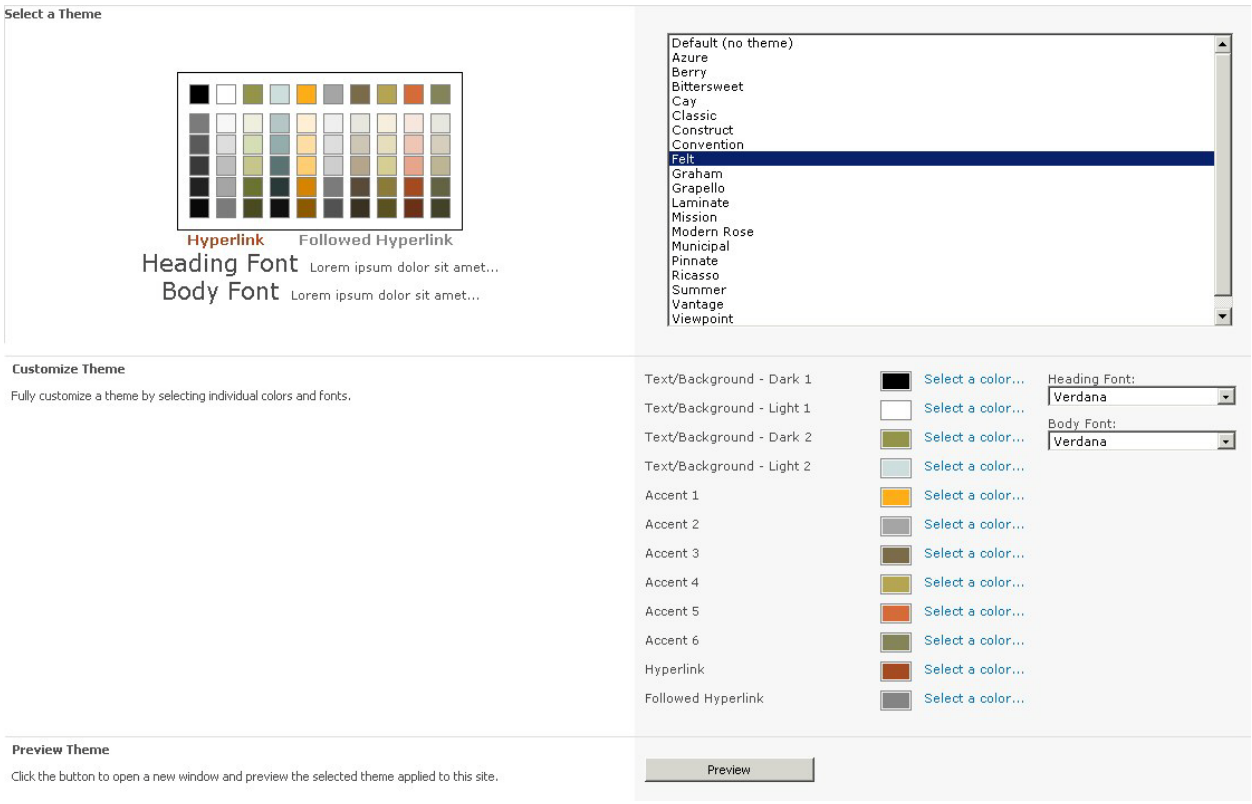

Figure 28: Theme in SharePoint Server 2010

SharePoint Designer

SharePoint Designer is a powerful tool for customizing SharePoint sites and adding functionality. With any tool deployed in an organization, applying an effective governance plan is key to ensuring responsible use of the tool. SharePoint Server 2010 gives administrators added control over how SharePoint Designer is used in each Web application. Administrators can control the following options:

- Enable SharePoint Designer: Controls whether SharePoint Designer can be used at all.
- Enable Detaching Pages from the Site Definition: Determines whether pages can be detached from the site definition. (SharePoint Designer only allows customization of a page when the page is put into advanced editing mode. This will greatly reduce the number of pages that will be detached from the site definition.)
- Enable Customizing Master Pages and Layout Pages: Controls whether site administrators are allowed to customize master pages and layout pages.

- Enable Managing of the Web Site URL Structure: Specifies whether site administrators can manage the URL structure of their site.

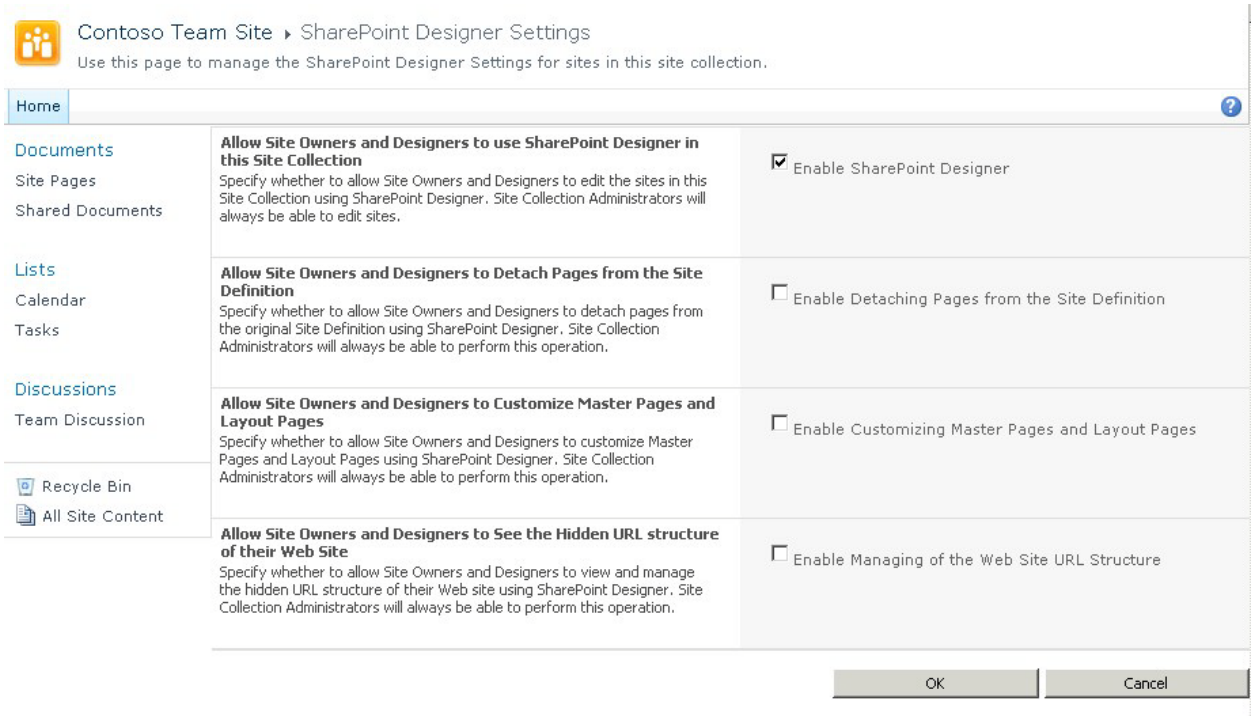
Contoso Team Site ▸ SharePoint Designer Settings
Use this page to manage the SharePoint Designer Settings for sites in this site collection.
Home
Documents
Site Pages
Shared Documents
Lists
Calendar
Tasks
Discussions
Team Discussion
Recycle Bin
All Site Content
Allow Site Owners and Designers to use SharePoint Designer in this Site Collection
Specify whether to allow Site Owners and Designers to edit the sites in this Site Collection using SharePoint Designer. Site Collection Administrators will always be able to edit sites.
Enable SharePoint Designer
Allow Site Owners and Designers to Detach Pages from the Site Definition
Specify whether to allow Site Owners and Designers to detach pages from the original Site Definition using SharePoint Designer. Site Collection Administrators will always be able to perform this operation.
Enable Detaching Pages from the Site Definition
Allow Site Owners and Designers to Customize Master Pages and Layout Pages
Specify whether to allow Site Owners and Designers to customize Master Pages and Layout Pages using SharePoint Designer. Site Collection Administrators will always be able to perform this operation.
Enable Customizing Master Pages and Layout Pages
Allow Site Owners and Designers to See the Hidden URL structure of their Web Site
Specify whether to allow Site Owners and Designers to view and manage the hidden URL structure of their Web site using SharePoint Designer. Site Collection Administrators will always be able to perform this operation.
Enable Managing of the Web Site URL Structure
OK
Cancel

Figure 29: SharePoint Designer Settings

Master Pages

While SharePoint Server 2010 still relies on master pages for implementing a custom look and feel to sites, there have been some enhancements made. Through the Web interface or by using SharePoint Designer, a site owner will be able to apply branding to their site, independent of other sites. This allows the administration of look and feel to be delegated as appropriate. For branding of application and system pages, this role will fall to the server administrator.

Administrators will be able to specify whether the system pages in the _Layouts folder are rendered by using the site master pages provided by site owners or by default master pages available across the system. In Office SharePoint Server 2007, pages rendered from this directory used application.master. This presented a challenge to organizations who wanted to create a custom user interface – because application.master is a system file and there was no option to option to use a custom master page for this. The only options available were to modify the system file or to

style the page by using a custom theme. It should be noted that customizing application.master was not recommended because in the event that something unfortunate happened to this master page, none of the system settings for a site could be accessed. Not only does SharePoint Server 2010 add greater flexibility for how to apply branding to these system pages, it also provides a fail-safe mechanism. If there is an error in the master page used for pages in the _Layouts, SharePoint will reference the default.master file so that system pages can still be accessed. Additionally, it is now possible to use Windows PowerShell to specify a customer master page to system error pages, login pages, confirmation pages, and other non–site-specific pages. By combining all of these options, creating a consistent brand across SharePoint is now easier than ever.

Silverlight

Microsoft Silverlight continues to become more important as a way to provide rich data to users. SharePoint Server 2010 provides a Web Part that allows users to quickly integrate their Silverlight applications into SharePoint Server 2010. This makes it easier for organizations to use SharePoint Server 2010 to store data and use Silverlight to create dashboards and other applications to provide tools to report and interact with critical data. The Silverlight Web Part will issue a token to the clients which will then pass that token to the Silverlight server. The server will then use this token to request data from the server running SharePoint Server directly. This token passing process will allow the Silverlight interactivity to remain secure, because SharePoint Server 2010 requires the token to return data to the Silverlight server, while remaining efficient because the server running SharePoint Server will not have to proxy the connections.

Conclusion

At its core, the principal uses for SharePoint® Products and Technologies (site provisioning, Web Parts, collaboration, document management, Web management, and so on) haven't changed. What has changed is the amount of out-of-the-box functionality that can be leveraged within an organization that has implemented this solution.

From a business perspective, SharePoint Server 2010 solves the need for a centralized solution that can be used in conjunction with existing software and applications. By leveraging Microsoft Business Connectivity Services, SharePoint Server 2010 can natively

connect to existing sources of information. With the multi-tenancy feature, organizations have more control with hosted environments (from the hosting side of things in addition to the client side).

For an administrator, this release has made drastic improvements toward a solution that is not only manageable, but is easy to maintain. The integration with Windows PowerShell™ allows for more administrative control, a more efficient command line, and improves productivity. The new Central Administration interface makes it easier to locate and configure settings for your environment. SharePoint Server 2010 has incorporated more performance controls and logging and reporting options, in addition to a much desired improvement on the native disaster-recovery options. The revamped service applications feature ensures greater flexibility with regards to the shared services being used within the environment.

Lastly, even the user has a better experience. The user interface is more closely related to Office applications, an interface with which users today are very familiar. The introduction of claims-based authentication allows for a better user experience as organizations incorporate a variety of user applications and Web-based solutions. And getting back to some of the business advantages, a centralized solution is a one-stop shop for users who need to access information related to their day-to-day activities.

As you can see, with the native capabilities available with SharePoint Server 2010, it is easy to understand the level of anticipation that has led up to this release.